BEST

Basic Envisat SAR Toolbox

User Manual

Version 4.0.2., March 2005

Please report bugs to eohelp@esa.int
Find further BEST information at http://envisat.esa.int/best/

This User Manual is a work-in-progress

You are invited to check the website regularly for updated versions

Contents

A OVERVIEW 4

1. Introduction 5
2. Three Simple Examples 8
3. BEST Functions Summary 12
4. BEST File Extensions and Internal Format 16
5. Installation 18
6. HMI functionality 23

B TOOLS 25

7. Data Import and Quick Look 26
   - Header Analysis 28
   - Media Analysis 33
   - Quick Look Generation 35
   - Full Resolution Extraction 40
   - Portion Extraction 43
   - Image Preview 45
   - Coordinates Retrieving by Example Image 46
   - Support Data Ingestion 48
   - Import GeoTIFF 50
   - Import TIFF 51
   - Import Raster Image 52
   - New Product Adding 55
8. Data Export 63
   - Export GeoTIFF 64
   - Export to TIFF 65
   - Export to BIL 67
   - Export to RGB 69
9. Data Conversion 70
   - Gain Conversion 71
   - Power to Amplitude Conversion 76
   - Amplitude to Power Conversion 77
   - Linear to dB Conversion 78
   - Complex to Amplitude Conversion 79
   - Integer to Float Conversion 80
   - Ancillary Data Dump 81
   - Image Operation 83
   - Geometric Conversion 87
   - Slant Range to Ground Range Conversion 91
   - Flip Image 94
   - Sensitivity Vector Evaluation 96

10. Statistical....98
Global Statistic....99
Local Statistic....101
Principal Components Analysis....104
11. Resampling....105
Oversampling (Up-Sampling)....106
Undersampling (Down-Sampling)....108
12. Co-registration and Coherence Generation....113
Co-registration....114
Coherence Generation....124
Footprint Registration....126
Image Geo-correction....128
Amplitude-Coherence Multi-layer Composite....132
13. Speckle Filter....135
Speckle Filter....136
14. Calibration....143
Backscattering Image Generation (ERS)....144
ADC Compensation (ERS)....148
Gamma Image Generation (ERS)....150
Backscattering Image Generation (ASAR)....151
Image Retro-calibration (ASAR)....153
Rough Range Calibration (ASAR)....155
Swath Enhancement (ASAR)....156

C APPENDICES....158

A OVERVIEW

1. Introduction

What is BEST?

The Basic Envisat SAR Toolbox (BEST) is a collection of executable software tools that has been designed to facilitate the use of ESA SAR data. The purpose of the Toolbox is not to duplicate existing commercial packages, but to complement them with functions dedicated to the handling of SAR products obtained from ASAR (Advanced Synthetic Aperture Radar) and AMI (Active Microwave Instrument) onboard Envisat and ERS 1&2 respectively. BEST has evolved from the ERS SAR Toolbox.

The Toolbox operates according to user-generated parameter files. The software is designed with an optional graphical interface that simplifies specification of the required processing parameters for each tool and (for Windows™ versions only) sets it running.

The interface doesn't include a display function. However, it includes a facility to convert images to TIFF or GeoTIFF format so that they can be read by many commonly available visualisation tools. Data may also be exported in the BIL format for ingestion into other image processing software.

The tools are designed to achieve the following functions:

Data Import and Quick Look: basic tools for extraction of data from standard format ESA SAR products, generation of quick look images, import of TIFF and GeoTIFF files and generic raster data.

Data Export: output of data to selected common formats, generation of RGB composites.

Data Conversion: conversion between different image formats, transformation of data by flipping or slant range to ground range re-projection, calculation of sensitivity vectors.

Statistical: calculation of global or local statistical parameters from real image data, computation of the principal components of multiple images.

Resampling: over and under sampling of an image by means of spatial and spectral methods.

Co-registration: automatic co-registration of two or more real or complex images (including ERS/Envisat pairs), evaluation of quality parameters, geometric correction of medium resolution products.

Support for Interferometry: computation of orbital baseline from DORIS files, calculation of interferometric coherence, evaluation of altitude of ambiguity.

Speckle Filtering: removal of speckle noise from a backscatter image.

Calibration: radiometric correction of Envisat and ERS images including retro-calibration of ASAR products and wide-swath image refinement.

Running BEST

The algorithms of the Toolbox are executed by means of the Human Machine Interface (HMI). Users are able to specify parameters, select input files and name output files according to the selected algorithm.

For Windows™ users there is a familiar Visual Basic interface. The HMI for LinuX and Solaris2™ users is written in Tcl (Tool Command Language). The Tcl/Tk software must be installed prior to running BEST on these platforms.

Both HMIs essentially automate the generation and execution of ASCII ".ini" files that are required by the Toolbox. However, it is perfectly possible to use the Toolbox without an HMI.

Some users may prefer to produce their own "*.ini*" files or edit existing ones to meet their specific needs and run these directly from the command prompt. To execute a tool, type the command:

```
BEST file_name.ini
```

where "*file_name.ini*" is an ASCII file containing the parameters necessary for a tool's execution.

For processing data using a series of tools, it is possible to edit "*.ini*" files together into a macro "*.ini*" file so that the entire procedure may be executed by a single command.

Later in this section, three simple examples are presented which describe in detail the various parameters of "*.ini*" files required to run some basic Toolbox functions.

What data can be read?

The Toolbox has been designed to handle ESA data products from both the Envisat ASAR instrument and the AMIs on ERS 1&2.

ASAR data acquired in Image Mode, Wide Swath Mode, Alternating Polarization Mode and Global Monitoring Mode, processed to Level 1b (SLC, Precision, Medium Resolution or Ellipsoid Geo-coded), is supported.

Image Data from ERS SAR, processed as RAW, SLC, SLCI, PRI, GEC or GTC, is also supported.

The Toolbox handles the standard Envisat product file format. For ERS data, products generated within the ESA ERS Ground Segment at D-PAF, I-PAF, UK-PAF and ESRIN are supported, plus data from many of the "foreign" stations in the following formats:

- ESA CEOS version 3.0, used by all ESA PAFs since January 1997.
- ESA CEOS version 2.1, used by ESA PAFs from October 1995 to January 1997, also used by several foreign stations, e.g. China, South Africa, Argentina, Singapore.
- ESA CEOS version 2.0, used by several foreign stations, e.g. Ecuador.

Toolbox formats and file extensions

The majority of Toolbox functions operate on data that has been converted into the Toolbox internal format. Therefore it is always necessary to first read new data into the Toolbox format using the Data Import tools (see Chapter 7). All Toolbox operations produce output data in the internal format and assign filename extensions that identify the tool used and the data type (see Chapter 4).

2. Three Simple Examples

The purpose of this chapter is to provide three simple examples of the most basic BEST functions. Hopefully this will help to demonstrate the way in which the Toolbox works, so that you can use it more effectively according to your own needs. In these examples, header information is read from the data, a quick look image is generated and a portion of the data is read onto disk.

Header Analysis

Before any processing can be performed on data using BEST (including quick look generation or data extraction), the HEADER ANALYSIS module must be run to extract into an internal format file the header information contained in the product or accompanying file.

The ASCII "*.ini*" file generated to run the tool may look something like this:

```
[HEADER ANALYSIS]
Output Dir = "C:\BEST_out\"
Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"
Input Media Type = "cdrom"
Sensor Id = "ASAR"
Sensor Mode = "Image"
Product Type = "PRI"
Data Format = "ENVISAT"
Source Id = "esp"
Number Of Volumes = 1
Annotation File = "header_IMP"
Header Analysis File = "header_IMP"
Dismount Volume = 'N'
```

Supposing the file is called "*header_analysis.ini*", the tool would be run using the command:

```
BEST header_analysis.ini
```

It is useful to examine the contents of the file "*header_analysis.ini*" to understand the meaning of the various instructions. Many further details about the options available for the HEADER ANALYSIS tool can be found in the main section of the User Manual.

`[HEADER ANALYSIS]`	*This is the name of the function.*
`Output Dir = "C:\BEST_out\"`	*This indicates path to a directory where the output files will be written.*
`Input Media Path = "D:\data\ASAR..."`	*This path directs the tool to the device and the product to be analysed. In this case it is a CD drive mounted on the D: drive.*
`Input Media Type = "cdrom"`	*The medium on which the data is held. In this case a CD-ROM from an ESA PAF.*
`Sensor Id = "ASAR"`	*The instrument or platform that acquired the data.*
`Sensor Mode = "Image"`	*For ASAR images, the mode in which the data was acquired. In this case it is Image Mode.*
`Product Type = "PRI"`	*The level to which the data is processed by the PAF.*

`Data Format = “ENVISAT”`	*The data format.*
`Source Id = “esp”`	*The ‘PAF’ at which the data was processed. This is relevant for ERS data; for Envisat products (as in this case) “esp” is always used to indicate ESRIN.*
`Number Of Volumes = 1`	*The number of tapes. This will usually be “1” unless the data is contained on more than 1 Exabyte tape.*
`Annotation File = “header_IMP”`	*The name of the output text file. This will automatically be given the extension “.txt”.*
`Header Analysis File = “header_IMP”`	*The name of the output Toolbox format file (input for many other function). This will be given the extension “.HAN”.*
`Dismount Volume = ‘N’`	*(This indicates that the volume drive would not be dismounted after the operation had finished.)*

Quick Look

The QUICK LOOK tool generates, directly from the original product, a TIFF file of selectable size showing a subsampled approximation of the detected SAR scene.

The ASCII “*.ini*” file generated to run the tool may look something like this:

```
[QUICK LOOK]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"
Input Media Type = "cdrom"
Header Analysis File = “header_IMP.HAN"
Output Quick Look Image= "ql_IMP"
Output Grid Image = "qlg_IMP"
Quick Look Presentation = "GEOGRAPHIC"
Number of Grid Lines = 2, 2
Output Image Size = 800, 0
Window Sizes = 3, 3
Grid Type = "LATLON"
Grid Drawing Mode = "transparent"
Min Percentage = 1
Max Percentage = 99
Dismount Volume = 'N'
```

Supposing the file is called “*quick_look.ini*”, the tool would be run using the command:

```
BEST quick_look.ini
```

It is useful to examine the contents of the file “*quick_look.ini*” to understand the meaning of the various instructions. Many further details about the options available for the QUICK LOOK GENERATION tool can be found in the main section of the User Manual.

`[QUICK LOOK]`	*This is the name of the function.*
`Input Dir = "C:\BEST_out\"`	*The path to the directory containing the required input files, in this case the header file “header_IMP.HAN”.*

`Output Dir = "C:\BEST_out\"`	*The path to a directory where the output files will be wrtitten.*
`Input Media Path = "D:\data\ASAR..."`	*This path directs the tool to the device and the product to be analysed. In this case it is a CD drive mounted on the D: drive.*
`Input Media Type = "cdrom"`	*The medium on which the data is held.*
`Header Analysis File = "header_IMP.HAN"`	*The required input file for this function, which contains information about the data product and was created by the HEADER ANALYSIS function.*
`Output Quick Look Image = "ql_IMP"`	*The name of the output image file. This will be in standard TIFF format with the extension ".tif" added.*
`Output Grid Image = "qlg_IMP"`	*As above. This version of the image has a grid superimposed on it. The extension ".tif" will be added.*
`Quick Look Presentation = "GEOGRAPHIC"`	*The orientation of the image in the output files. "Geographic" forces the data to be flipped so that North is at the top and East is to the right.*
`Number Of Grid Lines = 2, 2`	*The number of grid lines to be superimposed on the grid image in vertical and horizontal directions.*
`Output Image Size = 800, 0`	*The size of the output image in rows and columns. In this case the output will have 800 rows and squared pixels – the software will compute (and return in verbose) the necessary number of columns.*
`Window Sizes = 3, 3`	*The size of the window used to average the full resolution image to obtain the quick look image.*
`Grid Type = "LATLON"`	*The grid image will be annotated with lines of equal latitude and longitude.*
`Grid Drawing Mode = "transparent"`	*The labels on the grid image will not obscure the underlying image.*
`Dismount Volume = 'N'`	*(This indicates that the volume drive would not be dismounted after the operation had finished.)*

Full Resolution Extraction

The FULL RESOLUTION EXTRACTION tool reads data from the original product into the BEST internal format. It is a prerequisite for all subsequent processing. The user may opt to extract an entire scene or just a portion of it.

The ASCII "*.ini*" file generated to run the tool may look something like this:

```
[FULL RESOLUTION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"
Input Media Type = "cdrom"
Header Analysis File = "header_IMP.HAN"
Output Image = "full_res_IMP"
Coordinate System = "LATLON"
Centre = 52.406, 4.470
Size Unit = "KM"
Size = 3.1, 6.3
```

Supposing the file is called "*full_res.ini*", the tool would be run using the command:

```
BEST full_res.ini
```

It is useful to examine the contents of the file "*full_res.ini*" to understand the meaning of the various instructions. Many further details about the options available for the FULL RESOLUTION EXTRACTION tool can be found in the main section of the User Manual.

`[FULL RESOLUTION]`	*This is the name of the function.*
`Input Dir = "C:\BEST_out\"`	*The path to the directory containing the required input files, in this case the header file "header_IMP.HAN".*
`Output Dir = "C:\BEST_out\"`	*The path to a directory where the output files will be wrtitten.*
`Input Media Path = "D:\data\ASAR..."`	*This path directs the tool to the device and the product to be analysed. In this case it is a CD drive mounted on the D: drive.*
`Input Media Type = "cdrom"`	*The medium on which the data is held.*
`Header Analysis File = "header_IMP.HAN"`	*The required input file for this function, which contains information about the data product and was created by the HEADER ANALYSIS function.*
`Output Image = "full_res_IMP"`	*The name of the output file, which will be in the Toolbox internal format and which will be given the extension ".XTs" if the input image is PRI data (as in this case) or ".XTt" if the input image is SLC data.*
`Coordinate System = "LATLON"`	*The coordinate system used to define a subset of the data set for extraction. In this case, the location of the region of interest is identified by latitude and longitude (the coordinates might be derived from the superimposed grid on the quick look image, generated previously).*
`Centre = 52.406, 4.470`	*The location of the region of interest, defined, in this case, by the coordinates at its centre (given in decimal degrees).*
`Size Unit = "KM"`	*The system of units used to define the size of the region of interest to be extracted. In this case kilometres.*
`Size = 3.1, 6.3`	*The size of the region of interest (given in km).*

The output from the Full Resolution Extraction function (i.e. "*full_res_IMP.XTs*") may be viewed either as a quick look image, or by exporting to TIFF after first applying the GAIN CONVERSION tool to adjust the dynamic range of the pixel values and convert the data to 8 bits.

3. BEST Functions Summary

This chapter contains a brief summary of all the BEST functions.

Data Import and Quick Look tools

1. Header Analysis
Decodes the product header and stores the information in an internal Toolbox format file necessary for input to the FULL RESOLUTION EXTRACTION and QUICK LOOK GENERATION tools. Also writes the header information to an ASCII text file for reference purposes.

2. Media Analysis
Determines the number of files in each volume, the number of records in each file and the number of bytes in each record for products held on Exabyte media.

3. Quick Look Generation
Generates a reduced-resolution approximation of an image directly from the original data product or from an internal format file.

4. Full Resolution Extraction
Extracts a full resolution portion of an original data product to the internal file format.

5. Portion Extraction
Extracts a full resolution subset of an image already in the Toolbox internal format.

6. Image Preview
Extracts a region of interest from a quick look image. This function is useful to verify that a region of interest is correctly defined before it is extracted at full resolution.

7. Coordinates Retrieving by Example Image
Derives the coordinates within a scene that define a subset or region of interest, as extracted from a quick look image and saved as a second ".tif" file using another image viewing tool.

8. Support Data Ingestion
Converts support data (e.g. antenna pattern information or lookup tables for calibration) from an ESA ASCII format into the Toolbox internal format.

9. Import GeoTIFF
Converts a GeoTIFF image into the Toolbox internal format.

10. Import TIFF
Converts standard TIFF files to the Toolbox internal format.

11. Import Raster Image
Converts an image in raster format into the Toolbox internal format without having to specify the number of file header bytes or line header bytes. Also generates an ASCII file containing the image size information, which is compatible with the ERMAPPER ".ers" format.

Data Export tools

1. Export GeoTIFF
Converts data from internal format to a GeoTIFF image that includes geographic information.

2. Export to TIFF
Converts 8 bit data from the Toolbox internal format to standard TIFF format as either single-channel greyscale or 3-channel colour images.

3. Export to BIL
Converts one or more (up to ten) internal Toolbox format images having the same size and data type to one binary image in BIL (Band Interleaved by Line) format.

4. Export to RGB
Converts three internal Toolbox format images with the same size to a 24-bit RGB image.

Data Conversion tools

1. Gain Conversion
Rescales floating-point or real 16-bit integer data to 8 bits, thereby preparing it for export to formats that can be visualised in basic graphics packages.

2. Power to Amplitude Conversion
Converts a power image into an amplitude image.

3. Amplitude to Power Conversion
Converts an amplitude image into a power image.

4. Linear to dB Conversion
Converts an amplitude or intensity image with a linear scale into an image in decibel (dB) units.

5. Complex to Amplitude Conversion
Derives the amplitude modulus from a complex image.

6. Integer to Float Conversion
Converts a real image from the integer format to the floating-point format.

7. Ancillary Data Dump
Generates an ASCII listing of the image annotations relating to an image in the Toolbox internal format.

8. Image Operation
Performs basic algebraic operations (sum, subtract, multiply or divide) between two images or between one image and a constant factor. It is also possible to calculate the absolute value of a single image.

9. Geometric Conversion
Converts between row, column and latitude, longitude coordinates for points specified in any given image. Also calculates the satellite's position and angles of incidence and look for the specified points.

10. Slant Range to Ground Range Conversion

Reprojects images from slant range (range spacing proportional to echo delay) to ground range (range spacing proportional to distance from nadir along a predetermined ellipsoid). The tool works on complex data (extracted and/or co-registered SLC products) and real data (coherence products).

11. Flip Image
Executes a horizontal or vertical flip operation (or both) on any internal Toolbox format image.

12. Sensitivity Vector Evaluation
Calculates the sensitivity vector of an input image point by point.

Statistical tools

1. Global Statistic
Calculates a range of statistical parameters (mean, standard deviation, coefficient of variation, equivalent number of looks) for an image or region of interest within an image. Also generates a histogram of the pixel values.

2. Local Statistic
Generates output images showing a range of statistical parameters (mean, standard deviation, coefficient of variation, equivalent number of looks) computed from an image using a moving window of selectable size.

3. Principal Components Analysis
Generates the first and second principal components from a pair of input images.

Resampling tools

1. Oversampling (Up-Sampling)
Resamples an image to increase the number of pixels.

2. Undersampling (Down-Sampling)
Resamples an image to reduce the number of pixels.

Co-registration and Coherence Generation tools

1. Co-registration
Registers one or more images to another using up to three separate processes to achieve a precise fit. Images can be real or complex.

2. Coherence Generation
Calculates the phase coherence between two co-registered complex images.

3. Footprint Registration
Indicates on a quick look of a master image the ‘footprints’ of up to 10 co-registered slaves.

4. Image Geo-correction
Reprojects ASAR medium resolution imagery to a UTM or UPS planar grid.

5. Amplitude-Coherence Multi-layer Composite
Generates a multi-layer pseudo-true-colour composite image consisting of the coherence between two co-registered images with either their mean backscatter and the backscatter difference or the detected images of the master and slave.

Speckle Filtering tool

1. Speckle Filter
Removes speckle noise from real intensity images using the 'Gamma MAP' algorithm.

Calibration tools

For ERS data:

1. Backscattering Image Generation
Converts a power image into a backscatter image.

2. ADC Compensation
Corrects a power image for the ADC saturation phenomenon in ERS SAR products (prior to BACKSCATTERING IMAGE GENERATION).

3. Gamma Image Generation
Converts a backscatter image (i.e. output from BACKSCATTERING IMAGE GENERATION) into a Gamma image by dividing by the cosine of the incidence angle.

For ASAR data:

4. Backscattering Image Generation
Converts a power image into a backscatter image.

5. Retro-calibration
Removes an annotated antenna pattern and replaces it with another one.

6. Rough-range Calibration
Corrects ASAR Wide Swath and Global Monitoring Mode images for the effect of incidence angle variation from near to far range.

7. Enhancement Swath
Corrects ASAR Wide Swath and Global Monitoring Mode products affected by intensity discontinuities between sub-swaths

4. BEST File Extensions and Internal Format

The BEST output file extensions are designed to show which tool has created them and the type of data that they contain. The extension usually includes two upper case letters followed by a lower case letter. The upper case letters indicate the Toolbox function, e.g. PA = Power to Amplitude Conversion. The lower case letter indicates the format of the pixel data, following the convention:

i = 8-bit integer
s = 16-bit integer
t = complex integer, 16 bits + 16 bits
f = 32-bit float
c = complex float, 32 bits + 32 bits
r = RAW products, integer, 8 bits + 8 bits

Data Import and Quick Look:

Header Analysis	.HAN + .txt
Media Analysis	.txt
Quick Look Generation	.tif
Full Resolution Extraction	.XT?
Portion Extraction	.XT?
Image Preview	.tif
Coordinates Retrieving by Example	.txt
Support Data Ingestion	.SDf
Import GeoTIFF	.GT?
Import TIFF	.IT?
Import Raster Image (16-bit data)	.RIs
Import Raster Image (16+ 16-bit data)	.RIt

Data Export:

Export GeoTIFF	.tif
Export to TIFF	.tif
Export to BIL	.BG + .ers + .txt
Export to RGB	.tif

Data Conversion:

Gain Conversion	.GCi
Power to Amplitude Conversion	.PAf
Amplitude to Power Conversion	.APf
Linear to dB Conversion	.DBf
Complex to Amplitude Conversion	.CAf
Integer to Float Conversion	.IFf
Ancillary Data Dump	.txt
Image Operation	.OP?
Geometric Conversion	.txt
Slant to Ground Range Conversion	.SGf, .SGc
Flip Image	.FI?
Sensitivity Vector Evaluation	.txt

Statistical:

Global Statistic	.txt
Local Statistic	.LSf
Principal Component Analysis	.PCf

Resampling:

Oversampling (Up-Sampling)	.OV?
Undersampling (Down-Sampling)	.Unf

Co-registration and Coherence Generation:

Co-registration	.CR? + .XTf + .txt
Coherence Generation	.CHf
Footprint Registration	.tif
Image Geo-correction	.GRf
Amplitude-Coherence Composite	.tif

Radiometric Resolution Enhancement:

Speckle Filter	.SFf

Calibration:

Backscattering Image Generation	.BSf
ADC Compensation	.ADf
Gamma Image Generation	.GAf
Retro-calibration	.BSf
Rough Range Calibration	.XTf
Swath Enhancement	.XTf

N.B. "?" is replaced with the equivalent format indicator of the input data.

BEST Internal Format

The internal format adopted in BEST is called TTIFF, or Tiled Tagged Image File Format. TTIFF is a particular form of the commonly used TIFF format. The differences are essentially associated with the name of some image parameters (which, in the TIFF terminology, are called 'tags') and with some restrictions in the image organization. An extended discussion of this topic is given in Appendix 7.

The internal format TTIFF files can be read by standard display software packages (like XV for UNIX or ULEAD for PC), if the viewer supports the data type contained in the file. For example, it is possible to read 8-bit integer internal format images using XV. 8-bit integer images have the Toolbox file extension "*.??i*", where the question marks represent upper case letters indicating the module used to produce the image.

Of course, the EXPORT TO TIFF and EXPORT GEOTIFF tools allow any 8-bit Toolbox image to be converted to the standard TIFF format. Internal format data that is not 8-bit can be converted to 8-bit using the GAIN CONVERSION tool.

Important: When viewing a TIFF image generated by BEST (or an internal format file) using XV, it is necessary to launch the software first and load the image from the browser, rather than typing the command:

```
xv quicklook.tif
```

BEST data can also be exported using the EXPORT TO BIL tool. This converts one or more (maximum 10) integer or float images in the Toolbox internal format to a band interleaved by line (BIL) file (i.e. where consecutive records contain scan lines from each band in turn before moving from one row to the next) that can be used in an image viewer capable of ingesting such data (e.g. ERDAS or ER Mapper). Using the BIL format makes it possible to maintain the data in the source floating point representation, thereby retaining the accuracy of the data.

ER Mapper

ER Mapper includes an import function to load a TIFF image and transform it into its internal format. This option can also be activated via the operating system shell with the following command:

```
importmany TIFF-image-file ERMAPPER-image-file
```

Grey-level TIFF image files are transformed into a single-band ER Mapper file, while both RGB true-colour and palette-colour images are transformed into three-band ER Mapper image files.

5. Installation

Windows™ 98/2000/NT

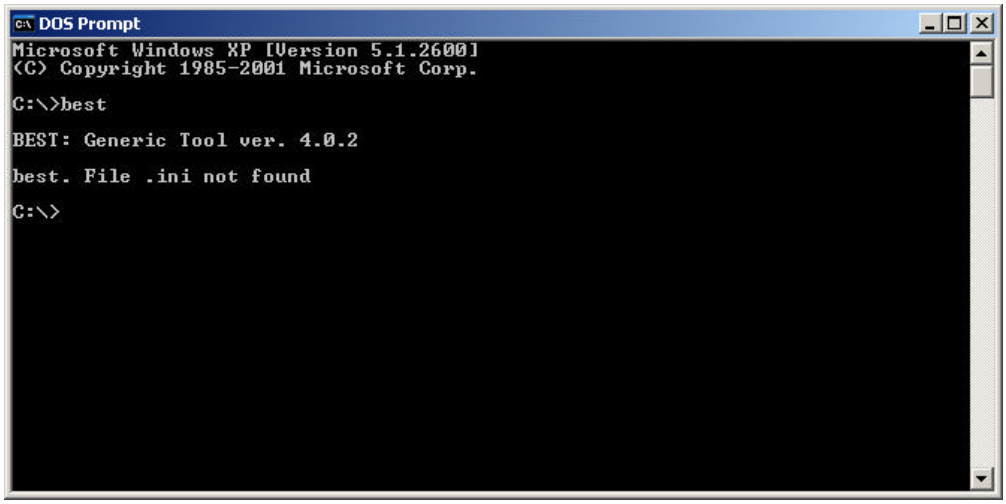
N.B. Windows™ 2000 or Windows™ NT users will need to be logged on as 'Administrator'.

1. Double-click the executable file and follow the instructions in the dialogue boxes.

N.B. The default destination folder is c:\best. Should the User prefer to install the software to the Program Files directory, it may be necessary to use the MS-DOS name PROGRA~1 in the path (at the Destination Folder page of the InstallShield dialogue). Paths containing spaces or with more than 8 characters are not handled by MS-DOS on the Windows™ 98 platform.

2. After restarting the computer (essential for Windows™ 98), check that the software is correctly installed by typing the command `BEST` in an MS-DOS window.

   If the software is correctly installed, you should see the following message:



3. The Visual Basic HMI is launched by double-clicking the BEST icon on the desktop (linked to "*c:\best\bin\BESTW.exe*").

Linux

1. It is first necessary to determine which shell will be used on the target system. The standard shell for Linux is the Bourne-Again shell, but the C shell, tcsh and the Korn shell are also possibilities. At the prompt in a newly created shell, type:

```
echo $SHELL
```

   The output indicates the current shell as follows:

```
/bin/csh   ⇒ the login shell is the C shell or tcsh
/bin/tcsh  ⇒ the login shell is tcsh
/bin/sh    ⇒ the login shell is the Bourne shell
/bin/bash  ⇒ the login shell is the Bourne-Again shell
/bin/ksh   ⇒ the login shell is the Korn shell
```

2. Create a home directory for BEST:

```
mkdir ~/BEST
```

3. Decompress the g-zipped tar file after moving it to the directory previously created:

```
tar xvfz software.tar.gz
```

   This will extract the ready-compiled BEST executables into the "*bin*" directory and the BEST shared library into the "*lib*" directory.

4a. If the login shell is the **C** shell or **tcsh** (see 1., above), modify or build the "*.cshrc*" file (found in the user's home directory) with the following lines:

```
setenv BESTHOME ~/BEST              ⇐ the home directory path; see 2., above
setenv FLAGFILE $BESTHOME/flagfile
setenv PATH $BESTHOME/bin:$PATH
```

4b. If the login shell is the **Bourne-Again** shell (see 1., above), modify or build the "*.bashrc*" file (found in the user's home directory) with the following lines:

```
BESTHOME=~/BEST                ⇐ the home directory path; see 2., above
FLAGFILE=$BESTHOME/flagfile
PATH=$BESTHOME/bin:$PATH
export BESTHOME FLAGFILE PATH
```

4c. If the login shell is the **Bourne** or **Korn** shell (see 1., above), modify or build the "*.profile*" file (found in the user's home directory) with the following lines:

```
BESTHOME=~/BEST                ⇐ the home directory path; see 2., above
FLAGFILE=$BESTHOME/flagfile
PATH=$BESTHOME/bin:$PATH
export BESTHOME FLAGFILE PATH
```

5. Exit from the current shell and create a new one.

   BEST is then ready to be run.

6. Check that the software is correctly installed by typing, at the prompt, the command:

```
best
```

   If the software is correctly installed, you should see the following message:

```
BEST: Generic Tool ver. 3.0 beta

best. File .ini not found
```

7. The Tcl/Tk HMI is launched by typing the command:

```
besthmi
```

   If you haven't already done so, you will need to download Tcl/Tk from the Tcl Developer Xchange (http://www.scriptics.com) and install it according to the accompanying instructions.

SunOS: Solaris2™

1. It is first necessary to determine which shell will be used on the target system. The default login shell for the SunOS is the Bourne shell, but the C shell and the Korn shell are also possibilities. At the prompt in a newly created shell, type:

```
echo $SHELL
```

   The output indicates the current shell as follows:

```
/bin/sh     ⇒ the login shell is the Bourne shell
/bin/csh    ⇒ the login shell is the C shell
/bin/ksh    ⇒ the login shell is the Korn shell
```

2. Create a home directory for BEST:

```
mkdir ~/BEST
```

3. Decompress the g-zipped tar file after moving it to the directory previously created:

```
tar xvfz software.tar.gz
```

   This will extract the ready-compiled BEST executables into the "*bin*" directory and the BEST shared library into the "*lib*" directory.

4a. If the login shell is the **C** shell (see 1., above), modify the "*.cshrc*" file (found in the user's home directory) with the following lines:

```
setenv BESTHOME ~/BEST              ⇐ the home directory path; see 2., above
setenv FLAGFILE $BESTHOME/flagfile
setenv PATH $BESTHOME/bin:$PATH
```

4b. If the login shell is the **Bourne** or **Korn** shell (see 1., above), modify the "*.profile*" file (found in the user's home directory) with the following lines:

```
BESTHOME=~/BEST                 ⇐ the home directory path; see 2., above
FLAGFILE=$BESTHOME/flagfile
PATH=$BESTHOME/bin:$PATH
export BESTHOME FLAGFILE PATH
```

5. Exit from the current session and re-login.

   BEST is then ready to be run.

6. Check that the software is correctly installed by typing, at the prompt, the command:

```
best
```

   If the software is correctly installed, you should see the following message:

```
BEST: Generic Tool ver. 3.0 beta

best. File .ini not found
```

7. The Tcl/Tk HMI is launched by typing the command:

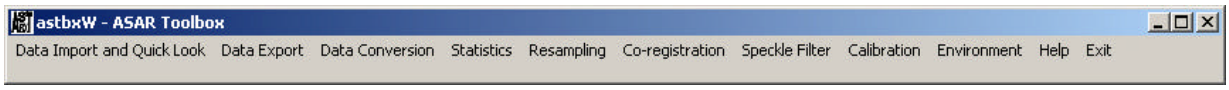
```
besthmi
```

If you haven’t already done so, you will need to download Tcl/Tk from the Tcl Developer Xchange (http://www.scriptics.com) and install it according to the accompanying instructions.

6. HMI functionality

The Visual Basic Human Machine Interface (for Windows™ versions) is launched by running the executable file *"c:\asartoolbox\bin\BESTW.exe"* (for example, by double-clicking its icon).

It consists of a set of menus that allow a dialogue box for each tool to be launched. The tools are arranged as they are in the body of this User Manual, according to the group to which they belong. In addition, there are menu groups for Environment, Help and Exit; some of the functions found here will be explained below.

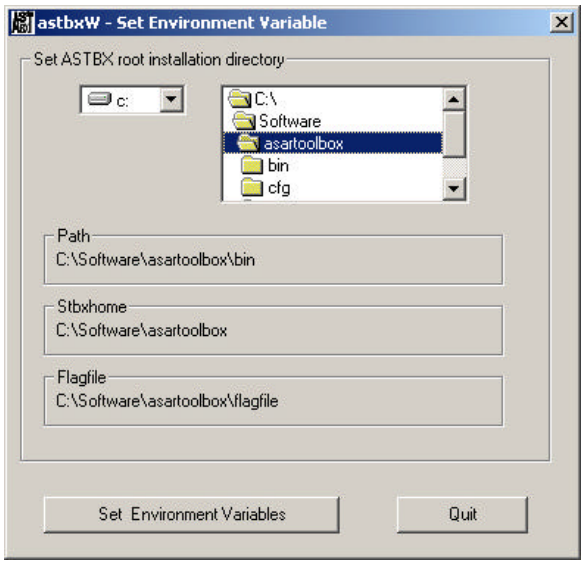


The Visual Basic HMI

In many of the dialogue boxes there is a [Show Default Values] button. This fills the fields in the dialogue box with typical or recommended values, which may then be altered if required. This is often a faster way to complete tool execution and reduces syntax errors.

Environment > Set Environment

Selecting Set Environment opens a dialogue box that allows the three environment variables required for installation to be set or reset quickly and easily.



Select the root installation directory by browsing in the directory tree in the upper part of the dialogue box and then click on [Set Environment Variables] to automatically complete the three environment variables and write them to the system settings. The resulting settings appear in the lower part of the dialogue box.

Help > Setup Working Directory

To ease the process of selecting input and output files from individual dialogue boxes, the default directory may be changed using this function at the beginning of a session. The specified

path (selected by browsing in a directory tree) is subsequently used as the value for ‘Input Dir’ and ‘Output Dir’ but, above all, the function enables the working files generated during the current processing session to be visible immediately when a dialogue box is opened, without first having to navigate to the correct directory. This makes file management on a large disk much easier.

The working directory is not retained between sessions but reverts to the specified PATH instead.

Exit

To close the ASAR Toolbox session, click on Exit > Exit. The working directory and parameters changed in any of the dialogue boxes will be reset.

B TOOLS

7. Data Import and Quick Look

This chapter documents the following tools:

1. Header Analysis
Decodes the product header and stores the information in an internal Toolbox format file necessary for input to the FULL RESOLUTION EXTRACTION and QUICK LOOK GENERATION tools. Also writes the header information to an ASCII text file for reference purposes.

2. Media Analysis
Determines the number of files in each volume, the number of records in each file and the number of bytes in each record for products held on Exabyte media.

3. Quick Look Generation
Generates a reduced-resolution approximation of an image directly from the original data product or from an internal format file.

4. Full Resolution Extraction
Extracts a full resolution portion of an original data product to the internal file format.

5. Portion Extraction
Extracts a full resolution subset of an image already in the Toolbox internal format.

6. Image Preview
Extracts a region of interest from a quick look image. This function is useful to verify that a region of interest is correctly defined before it is extracted at full resolution.

7. Coordinates Retrieving by Example Image
Derives the coordinates within a scene that define a subset or region of interest, as extracted from a quick look image and saved as a second ".tif" file using another image viewing tool.

8. Support Data Ingestion
Converts support data (e.g. antenna pattern information or lookup tables for calibration) from an ESA ASCII format into the Toolbox internal format.

9. Import GeoTIFF
Converts a GeoTIFF image into the Toolbox internal format.

10. Import TIFF
Converts standard TIFF files to the Toolbox internal format.

11. Import Raster Image
Converts an image in raster format into the Toolbox internal format without having to specify the number of file header bytes or line header bytes. Also generates an ASCII file containing the image size information, which is compatible with the ERMAPPER ".ers" format.

12. New Product Adding

This is not really a function in the same sense as the other tools. However, these pages describe how it is possible to recognise and decode SAR products that were not previously recognised as standard products.

Header Analysis

Description

The HEADER ANALYSIS function decodes all the header parameters from a product on tape, CD-ROM or hard disk. This information is extracted and stored in a plain ASCII file (extension .txt) and in a file in the Toolbox internal format (extension .HAN). The ASCII file can be examined using a standard text editor to provide useful information about the data. An example of one of these ASCII files is provided in Appendix 1.

The Toolbox has been designed to handle ESA data products from both the Envisat ASAR instrument and the AMIs on ERS 1&2. Level 1b ASAR data acquired in Image Mode, Wide Swath Mode, Alternating Polarization Mode or Global Monitoring Mode may be input, along with ERS image data (RAW, SLC, SLCI, PRI, GEC or GTC).

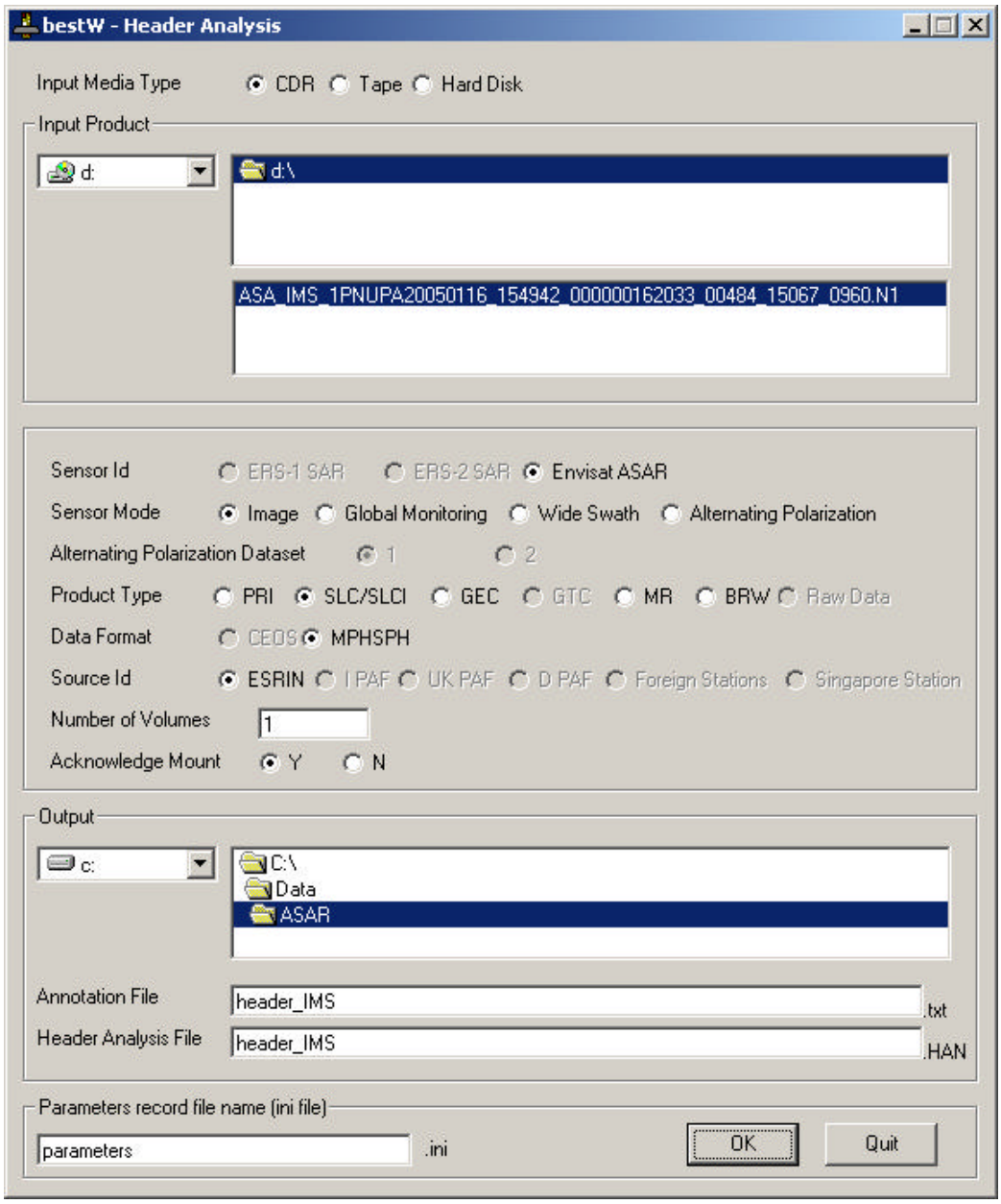
The Toolbox handles the standard Envisat product file format. For ERS data, products generated within the ESA ERS ground segment at D-PAF, I-PAF, UK-PAF and ESRIN are supported, plus data from non-ESA PAF stations, if they are delivered with ESA CEOS annotations; this is the case for the following SAR products:

- SAR products delivered by CRISP processor, located at Singapore station.
- SAR products delivered by ACS w-k processor located in Argentina (Cordoba), China (Beijing), Ecuador (Cotopaxi), Israel (Tel-Aviv), Kenya (Malindi), Russia, South Africa, Thailand (Bangkok).

The HEADER ANALYSIS module checks that images are generated from ESA products. This is done by testing that the log_vol_gen_agency tag is exactly “ESA”, except on Singapore products for which log_vol_gen_agency tag has to be exactly “CRISP”.

Important: The output file in the Toolbox internal format, which has the extension .HAN, is a necessary input to the FULL RESOLUTION EXTRACTION and QUICK LOOK GENERATION functions (unless ‘Input Media Type’ is set to “file” for the latter).

HMI



Typical HMI settings for reading an ASA_IMS_1P product

Notes:

Select the product by means of the ‘Input Media Path’ and ‘Input Product Image’ fields (note that the ‘Sensor Id’ must be specified before image products appear as selectable).

The ‘Sensor Mode’ field is enabled only for the Envisat ASAR sensor.

The ‘Alternating Polarization Dataset’ field is enabled only for ASAR AP products; it distinguishes between the 1st and 2nd MDS.

Product Type: “PRI” (Precision products: IMP, APP)
“MR” (Medium Resolution products: IMM, WSM, ...)
“SLC” (Complex products: IMS, APS)
“GEC” (Geocoded products: IMG, APG)

"BRW" (Browse products: IM__BP, AP__BP, ...)

The 'Number of Volumes' field is relevant for import from Exabyte tape only.

Typical Processing Chain

HEADER ANALYSIS ⇒ QUICK LOOK ⇒ FULL RESOLUTION

Example "INI" file

```
[HEADER ANALYSIS]
Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"
Input Media Type = "cdrom"
Sensor Id = "ASAR"
Sensor Mode = "Image"
Product Type = "PRI"
Data Format = "ENVISAT"
Source Id = "esp"
Number Of Volumes = 1
Output Dir = "C:\BEST_out\"
Annotation File = "header_IMP"
Header Analysis File = "header_IMP"
Dismount Volume = 'N'
```

Parameter Summary: Header Analysis

Input Media Path

The path of the media unit:

- for a PC CDROM use:
  `Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"`
- for a Unix EXABYTE device use:
  `Input Media Path = "/dev/rst1"`
- for a Unix CDROM device use the entire path to the selected scene (ERS SAR product CDROMs can have multiple scenes on them):
  `Input Media Path = "/cdcom/SCENE1/"`

mandatory INPUT

BEST extension: (data product)

Input Media Type

The source media of the product:

- "tape" (Exabyte)
- "cdrom"
- "disk" (hard disk)

Example: `Input Media Type = "cdrom"`

mandatory parameter

Sensor Id

The platform from which the data was acquired:

- “ers1”
- “ers2”
- “ASAR”

Example: `Sensor Id = "asar"`

mandatory parameter

Sensor Mode

The mode in which Envisat ASAR data was acquired:

- “Image” (IM)
- “Wide Swath” (WS)
- “Global Monitoring” (GM)
- “Alternating Polarization” (AP) (note spelling with a “z”)

Example: `Sensor Mode = “Image”`

mandatory parameter IF ‘Sensor Id’ is “ASAR”

AP Dataset

The channel of an Envisat ASAR Alternating Polarization product to process, selectable between MDS1 or MDS2.

Example: `AP Dataset = 1`

mandatory parameter IF ‘Sensor Id’ is “ASAR” AND ‘Sensor Mode’ is “Alternating Polarization”

Product Type

The type of data product:

- “PRI” (Precision products, IMP, APP)
- “MR” (Medium Resolution products: IMM, APM, WSM)
- “SLC” (Complex products, IMS, APS)
- “GEC” (Geocoded products: IMG, APG)
- “BRW” (Browse products: IM__BP, AP__BP, WS__BP, GM__BP)
- “RAW” (ERS SAR RAW products)

Example: `Product Type = "pri"`

mandatory parameter

Data Format

The format of the product:

- “ceos” (for ERS data)
- “Envisat” (for Envisat data in mphsph format)

Example: `Data Format = "envisat"`

mandatory parameter

Source Id

The PAF or station where the data was processed:

- “esp” (for all Envisat data and ERS data processed at ESRIN products)
- “dep” (for ERS data processed at D-PAF)
- “ukp” (for ERS data processed at UK-PAF)
- “itp” (for ERS data processed at I-PAF)
- “sis” (for ERS data processed at Singapore Station)
- “fst” (for ERS data processed by an ACS w-k processor in Argentina (Cordoba), China (Beijing), Ecuador (Cotopaxi), Israel (Tel-Aviv), Kenya (Malindi), Russia, South Africa or Thailand (Bangkok))

Example: `Source Id = "esp"`
mandatory parameter

Number Of Volumes

The number of Exabyte cassettes into which the entire product is subdivided (usually 1).
Example: `Number Of Volumes = 1`
mandatory parameter IF ‘Input Media Type’ is “tape”

Annotation File

The name to be given to a text file that will contain a listing of all the header parameters (an extension “*.txt*” is automatically added by the system).
Example: `Annotation File = "header_IMP"`
mandatory OUTPUT
BEST extension: “.txt”

Header Analysis File

The name to be given to an internal format file that will contain all the decoded annotations for use in subsequent processing (an extension “*.HAN*” is automatically added by the system).
Example: `Header Analysis File = "header_IMP"`
mandatory OUTPUT
BEST extension: “.HAN”

Dismount Volume

A flag indicating whether the media shall be dismounted from the unit at the end of the volume processing; shall be set to “N” when a series of repeated extraction operations are planned on the same cassette, thus avoiding repeated unit mounting. This parameter is ignored (i.e. is assumed “Y”) for multi volume processing.
Example: `Dismount Volume = 'N'`
optional parameter (default is “Y”)

Media Analysis

Description

The MEDIA ANALYSIS function determines from a product held on Exabyte tape the number of files in each volume, the number of records in each file and the number of bytes in each record.

Important: Media analysis is only possible for data on Exabyte; it will not work for data on CDROM.

The information extracted by the MEDIA ANALYSIS function is stored in a file called the Media Content Report (output MCR file) and can be used for the following two purposes:

1) The media content report contains a clear summary of the product's physical structure and can therefore be used to quickly check that the data on the tape corresponds to its label.

2) If a SAR product does not follow the foreseen CEOS structure (if it has come from an exotic PAF/Station or if it is damaged), media analysis will help the user to understand its condition and may provide the necessary information to customise a FDF file and thus read the data.

The product recognition operation relies on the correlation of the file structure of the media to a predefined model. In case of discrepancies, there is a risk of product misrecognition.

To make use of this function it is necessary to read the output ASCII MCR file and evaluate whether the product under consideration is damaged to a degree that makes it un-readable, or whether the unexpected format encountered can be incorporated within the Toolbox framework by the creation of a new FDF file.

Note: An example of an output ASCII MCR file is shown in Appendix 2.

Typical Processing Chain

MEDIA ANALYSIS ⇒ HEADER ANALYSIS ⇒ QUICK LOOK

Example "INI" file

```
[MEDIA ANALYSIS]
Input Media Path = "/dev/rst1"
Number Of Volumes = 1
Output Dir = "./"
Output MCR File = "mcr"
Header Analysis File = "header_IMP"
Dismount Volume = 'N'
```

Parameter Summary: Media Analysis

Input Media Path

The path of the Exabyte unit.
Example: `Input Media Path = "/dev/rst1"`
mandatory INPUT
BEST extension: not applicable (SAR data)

Number Of Volumes

The number of Exabyte cassettes on which the entire product is held (usually 1).
Example: `Number Of Volumes = 1`
mandatory parameter

Output MCR File

The name of the file which will contain the media content report (an extension "*.txt*" is automatically added by the system).
Example: `Output MCR File = "mcr"`
mandatory OUTPUT
BEST extension: ".txt"

Quick Look Generation

Description

The QUICK LOOK GENERATION function is used to generate a reduced resolution, standard TIFF format version of an image. This is done using averaging and sub-sampling operations on the full resolution data to enable the user to quickly inspect an image.

The full resolution data can be accessed directly from tape or CD-ROM (thus avoiding the creation of large temporary files on the local disk) or from any file that has been created in the Toolbox internal format (except for integer 8-bit files, i.e. type 'i', and those generated by this QUICK LOOK GENERATION function or the Data Export tools).

Important: When starting from an original product, the QUICK LOOK GENERATION function requires the Header Analysis File (extension "*.HAN*") previously generated on the same product, which will contain product identifier parameters needed to access the data from the media.

The size of the output image is user-defined. The software can, optionally, compute the length of one axis, given the length of the other, assuming 'square' pixels. In the case of multi-looked input data, this means maintaining the aspect ratio of the image. For single look data, the software performs nominal multi-looking in the azimuth direction unless both axes are constrained by the user.

The output image is generated in two forms, one 'clean' and the other with a grid superimposed to help locate a scene and retrieve coordinates for points within the image. The two coordinate systems in which the grid can be generated are: row, column and latitude, longitude.

Important: When starting from data in an internal format file, the data may or may not contain the required ancillary geolocation parameters. If these parameters are not present (this will be the case if the image is the output from the IMPORT RASTER IMAGE function of the Data Import tool), the grid can be drawn only in row, column coordinates.

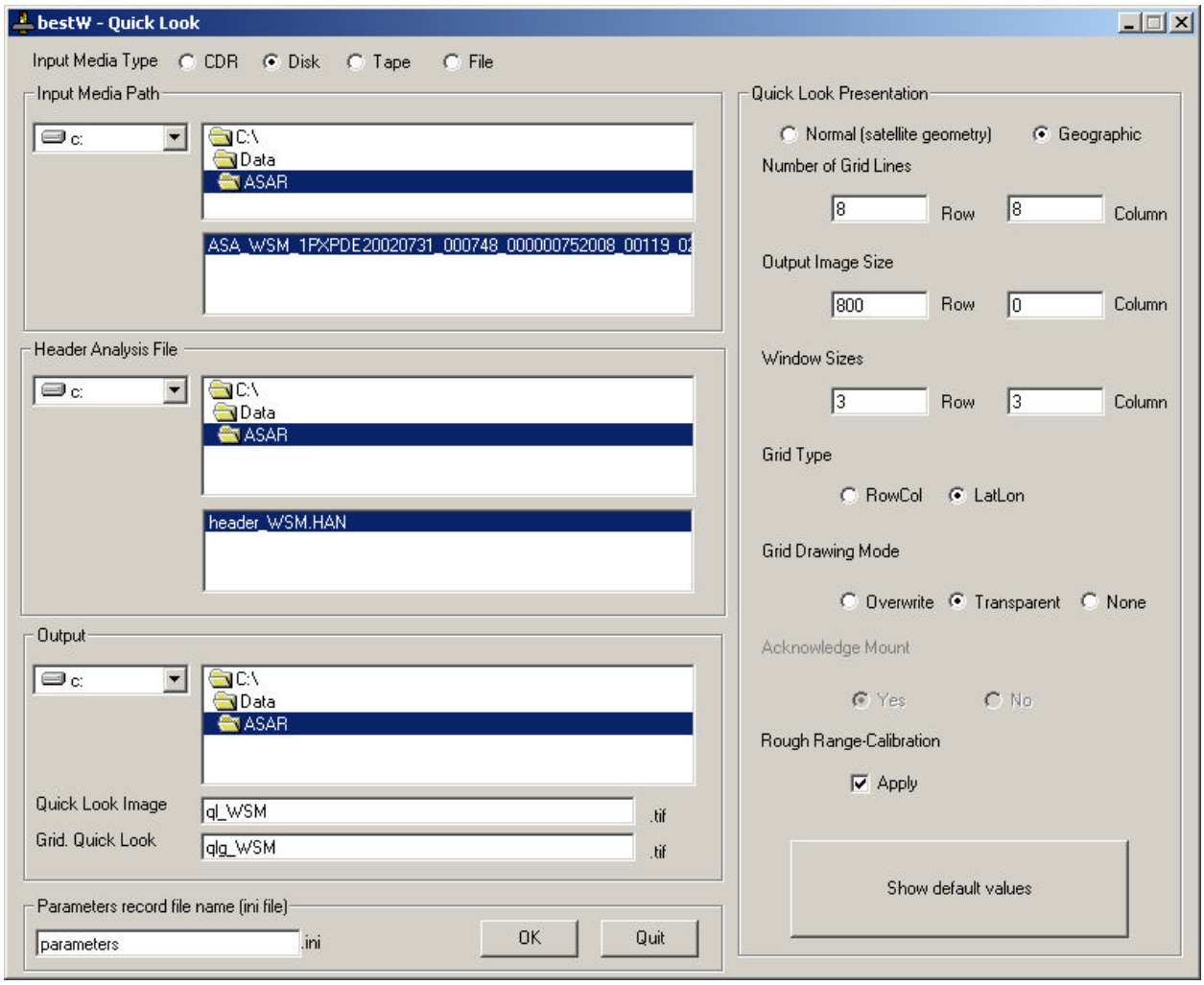
The quick look image can be displayed in a geometric orientation (option "GEOGRAPHIC", i.e. so that north is up, south is down, west is left and east is right) or in an orientation 'as viewed' by the satellite (option "NORMAL").

A rough range calibration may also be applied during the quick look generation to account for variation of incidence angle across the swath width. Whilst the aesthetic improvement is most noticeable in Wide Swath and Global Monitoring Mode products, the option is available for all ASAR and ERS data except geocoded products (i.e. ERS GEC and GTC, ASAR APG and IMG).

The output image is stored in standard TIFF format so it can be read using any TIFF reader (e.g. XV on Solaris2™, ulead on PC).

Important: It is not possible to open the TIFF files generated by BEST with all image viewing software. For PC platforms you should not encounter any problems using Adobe® Photoshop®, Jasc® Paint Shop Pro™ or Microsoft® Paint (a standard component of Microsoft Windows™ found in the Start Menu under Programs > Accessories > Paint). For Solaris2™ platforms using XV, it is necessary to launch the software first and then load the image from the browser.

HMI



Typical HMI settings for an ASA_WSM_1P product copied to the hard disk

Notes:

Select the product by means of the ‘Input Media Path’ and the ‘Header Analysis File’ (“*.HAN*”).

Typical Processing Chain

HEADER ANALYSIS ⇒ QUICK LOOK

Example "INI" file

```
[QUICK LOOK]
Input Media Path = "C:\Data\ASAR\ASA_WSM_1P ... 0053.N1"
Input Media Type = "disk"
Input Dir = " C:\Data\ASAR\"
Output Dir = " C:\Data\ASAR\"
Header Analysis File = “header_WSM.HAN"
Output Quick Look Image= "ql_WSM"
Output Grid Image = "qlg_WSM"
Quick Look Presentation = "GEOGRAPHIC"
Number of Grid Lines = 8 ,8
Output Image Size = 800 ,0
Window Sizes = 3 ,3
Grid Type = "LATLON"
Grid Drawing Mode = "transparent"
Min Percentage = 1
Max Percentage = 99
Rough Range-Calibration = "APPLY"
Dismount Volume = 'N'
```

Parameter Summary: Quick Look Generation

Input Media Type

The source media of the product:

- “tape” (Exabyte)
- “cdrom”
- “disk” (product on hard disk)
- “file” (BEST internal format)

Example: `Input Media Type = "cdrom"`

mandatory parameter

Input Media Path

The path of the media unit or, when ‘Input Media Type’ is set to “file”, the file name of the input internal format image.

- for a PC CDROM use:
  `Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"`
- for a Unix EXABYTE device use:
  `Input Media Path = "/dev/rst1"`
- for a Unix CDROM device use the entire path to the selected scene (ERS SAR product CDROMs can have multiple scenes on them):
  `Input Media Path = "/cdcom/SCENE1/"`

mandatory INPUT

BEST extension: not applicable IF ‘Input Media Type’ is “tape”, “cdrom” or “disk”
“.??f”, “.??c”, “.??s”, “.??t” IF ‘Input Media Type’ is “file”
where "??" indicates output from any BEST tool (except Data Export tools)

Header Analysis File

The internal format file containing all the decoded annotations, obtained during the HEADER ANALYSIS operation on the same product (with the associated extension ".*HAN*"). The parameter is ignored IF 'Input Media Type' is "file" (the header data comes from the internal image format annotations).

Example: `Header Analysis File = "header_WSM.HAN"`

mandatory INPUT IF 'Input Media Type' is "tape" or "cdrom"

BEST extension: ".HAN"

Output Quick Look Image

The name to be given to the standard TIFF file containing the quick look image, stretched to 8-bit and without a grid annotation (an extension "*.tif*" is automatically added by the system).

Example: `Output Quick Look Image = "ql_WSM"`

mandatory OUTPUT

BEST extension: ".tif"

Output Grid Image

The name to be given to the standard TIFF file containing the quick look image, stretched to 8-bit and annotated with a grid (an extension "*.tif*" is automatically added by the system).

Example: `Output Grid Image = "qlg_WSM"`

mandatory OUTPUT

BEST extension: ".tif"

Quick Look Presentation

The orientation of the output image:

- "GEOGRAPHIC" (with north at the top, south at the bottom, west to the left and east to the right)
- "NORMAL" (in an orientation "as viewed" by the satellite)

Example: `Quick Look Presentation = "GEOGRAPHIC"`

optional parameter (default is "GEOGRAPHIC")

Number Of Grid Lines

The number of iso-row or (iso-latitude) lines and iso-column (or iso-longitude) lines in the grid annotation; the first number refers to iso-row or iso-latitude lines; at least one of number shall be greater than zero

Example: `Number Of Grid Lines = 8, 8`

mandatory parameter

Output Image Size

The number of rows and columns in the output quick look image; the first number indicates the number of rows.

Example: `Output Image Size = 800, 800`

To maintain the aspect ratio of a multi-looked input image or perform nominal multi-looking on a single-look input image, set one of the values to "0". This invokes the system to compute an appropriate length for the second axis based on the single dimension defined.

To generate a quick look image of a multi-looked input with 500 rows and square pixels use:

```
Output Image Size = 500, 0
```

To generate a quick look image of a single-look input with 600 columns and nominal multi-looking in the azimuth direction use:

```
Output Image Size = 0, 600
```

mandatory parameter

Window Size

The number of rows and columns in the moving window used to average the full resolution data during the quick look creation; the first number indicates the number of rows.
Use “1” for a pure sub-sampling and a greater number to obtain a more smoothed image.
Example: `Window Size = 3, 3`
mandatory parameter

Grid Type

The type of grid lines to be used:
- “ROWCOL” (rows and columns)
- “LATLON” (latitude and longitude)

Example: `Grid Type = "LATLON"`
mandatory parameter

Grid Drawing Mode

The drawing style for the numerical grid labels:
- “overwrite” (gives the labels a black background)
- “transparent” (only the text itself obscures the underlying image)
- “none” (no labels are written on the image)

Example: `Grid Drawing Mode = "transparent"`
mandatory parameter

Rough Range Calibration

An optional flag to invoke approximate correction of intensity across the image swath caused by incidence angle variation.
Example: `Rough Range-Calibration = "APPLY"`
optional parameter (calibration only applied if present)

Acknowledge Mount

This parameter is used to avoid the request to acknowledge the unit mount during the quick look generation. To execute a header extraction immediately followed by a quick look generation (using a unique “*.ini*” file), set ‘Dismount Volume’ = “N” in the HEADER ANALYSIS module and set ‘Acknowledge Mount’ = “N” in the quick look module:

```
[HEADER ANALYSIS]
...
Dismount Volume = 'N'
[QUICK LOOK]
...
Acknowledge Mount = 'N'
```

This parameter is ignored (i.e. is assumed “Y”) for multi volume processing.
optional parameter (default is “Y”)

Dismount Volume

A flag indicating whether the media shall be dismounted from the unit at the end of the volume processing; shall be set to “N” when a series of repeated extraction operations are planned on the same cassette, thus avoiding repeated unit mounting. This parameter is ignored (i.e. is assumed “Y”) for multi volume processing.
Example: `Dismount Volume = 'N'`
optional parameter (default is “Y”)

Full Resolution Extraction

Description

The FULL RESOLUTION EXTRACTION function is used to extract a full resolution image portion from a product on tape, CD-ROM or hard disk.

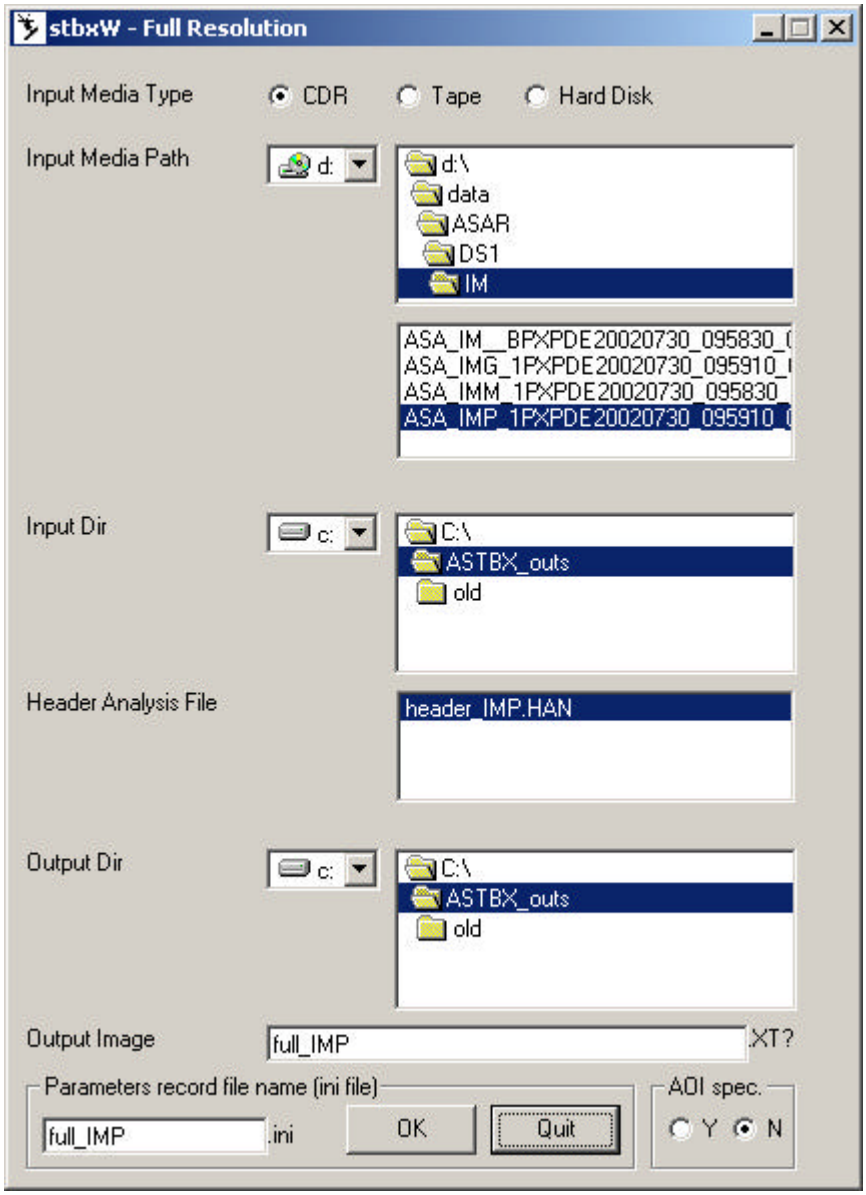
The resulting image file will be in the BEST internal format and will contain the image pixels plus the various header fields (i.e. the image ancillary data) already obtained with the HEADER ANALYSIS operation.

The extracted image has the same pixel format as the source data (no conversion is applied on the pixel values). Hence, the output image from the FULL RESOLUTION EXTRACTION tool will be given an extension "*.XT?*", where the question mark will be replaced by either r, i, s, t, f or c, depending on the data being read:

r when the operation takes place on ERS SAR RAW products from the source media
i when the operation takes place on 8-bit data generated by the gain conversion tool
s when the operation takes place on Precision or Geocoded products from the source media
t when the operation takes place on Complex products from the source media
f when the operation takes place on internal format data (not generated by gain conversion, oversampling complex data, co-registering complex data or importing raster data)
c when the operation takes place on internal format data (generated by oversampling complex data, co-registering complex data or importing raster data)

The image portion (also called AOI, area of interest) can be specified in all the methods described in Appendix 4.

HMI



Typical HMI settings for an ASA_IMP_1P product

Notes:

Select the product by means of the ‘Input Media Path’ and the ‘Header Analysis File’ (“*.HAN*”).

Typical Processing Chain

HEADER ANALYSIS ⇒ FULL RESOLUTION

Example "INI" file

```
[FULL RESOLUTION]
Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"
Input Media Type = "cdrom"
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Header Analysis File = "header_IMP.HAN"
Output Image = "full_IMP"
```

```
Top Left Corner = 0, 0
Bottom Right Corner = 511, 511
```

Parameter Summary: Full Resolution Extraction

Input Media Type

The source media of the product:

- "tape" (Exabyte)
- "cdrom"
- "disk" (hard disk)

Example: `Input Media Type = "cdrom"`
mandatory parameter

Input Media Path

The path of the media unit:

- for a PC CDROM use:
  `Input Media Path = "D:\data\ASAR\DS1\ASA_IMP_1P ... 320.N1"`
- for a Unix EXABYTE device use:
  `Input Media Path = "/dev/rst1"`
- for a Unix CDROM device use the entire path to the selected scene (ERS SAR product CDROMs can have multiple scenes on them):
  `Input Media Path = "/cdcom/SCENE1/"`

mandatory INPUT
BEST extension: (data product)

AOI specification

see Appendix 4
optional parameter (default is entire input image)

Header Analysis File

The internal format file containing all the decoded annotations, obtained during the HEADER ANALYSIS operation on the same product (with the associated extension "*.HAN*").
Example: `Header Analysis File = "header_IMP.HAN"`
mandatory INPUT
BEST extension: ".HAN"

Output Image

The name to be given to the internal format image that will contain the selected area of interest at full resolution (an extension "*.XT?*" is automatically added by the system, where the "?" indicates that the output image retains the same format as the input image).
Example: `Output Image = "full_IMP"`
mandatory OUTPUT
BEST extension: "XT?" where "?" indicates that the output image retains the same format as the input image.

Portion Extraction

Description

The PORTION EXTRACTION function extracts a full resolution sub-scene from an image already ingested into the Toolbox file format.

It is much faster to use the PORTION EXTRACTION tool to generate sub-scenes from data that is already in the BEST internal format, compared to extracting data directly from a tape or CD using the FULL RESOLUTION EXTRACTION function. It may therefore be of benefit, if the location of a feature is uncertain, to first use FULL RESOLUTION EXTRACTION to ingest a region of interest that is larger than necessary and subsequently identify and extract a smaller sub-scene using PORTION EXTRACTION. In this way it will only be necessary to use the relatively slow FULL RESOLUTION EXTRACTION function once.

The input image must be in the BEST internal file format and can be any size (it does not need to correspond to an entire full resolution data set). The area of interest (AOI) to be extracted can be specified in all of the methods described in Appendix 4, excluding the example image mode but including the polygonal AOI. In the latter case, pixel values outside the AOI are set to zero. When the input image does not contain the orbital and timing annotations (as in the case of images obtained with the IMPORT RASTER IMAGE function) the specification of the AOI using latitude and longitude is not possible.

Typical Processing Chain

HEADER ANALYSIS ⇒ FULL RESOLUTION EXTRACTION ⇒ PORTION EXTRACTION

Example "INI" file

```
[PORTION EXTRACTION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "fullres_data.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 511, 511
Output Image = "fullres_portion"
```

Parameter Summary: Portion Extraction

Input Image

The name of the input image in internal format
Example: `Input Image = "fullres_data.XTs"`
mandatory INPUT
BEST extension: “.??i”, “.??f”, “.??c”, “.??s”, “.??t”, “.??r” where "??" indicates that it is not important which BEST module produced the file.

AOI specification

See Appendix 4; the example image mode is not permitted and the latitude, longitude mode is permitted only if the orbital and timing information are present.
optional parameter (default is entire input image)

Output Image

The name of the image containing the image portion (an extension "*.XT?*" is automatically added by the system, where "?" indicates that the output image retains the same format as the input image).
Example: `Output Image = "fullres_portion"`
mandatory OUTPUT
BEST extension: ".XT?" where "?" indicates that the output image retains the same format as the input image.

Image Preview

Description

The IMAGE PREVIEW function extracts a region of interest from a quick look image (i.e. a "*.tif*" image generated using the QUICK LOOK GENERATION function). This function is useful to verify that the definition of an AOI is correct, before extracting the region from a full resolution image.

The output image is in the same standard TIFF format used for the quick look image.

Important: It is not possible to open the TIFF files generated by BEST with all image viewing software. For PC platforms you should not encounter any problems using Adobe® Photoshop®, Jasc® Paint Shop Pro™ or Microsoft® Paint (a standard component of Microsoft Windows™ found in the Start Menu under Programs > Accessories > Paint). For Solaris2™ platforms using XV, it is necessary to launch the software first and then load the image from the browser.

Typical Processing Chain

HEADER ANALYSIS ⇒ QUICK LOOK GENERATION ⇒ IMAGE PREVIEW ⇒ FULL RESOLUTION EXTRACTION

Example "INI" file

```
[IMAGE PREVIEW]
Input Image = "quicklook.tif"
Coordinate System = "ROWCOL"
Start Column = 100
Start Row = 100
End Column = 600
End Row = 600
Output Image = "preview"
```

Parameter Summary: Image Preview

Input Image

The name of the full quick look image; the version with or without a grid can be used.
Example: `Input Image = "quick look.tif"`
mandatory INPUT
BEST extension: "*.tif*"

AOI specification

See Appendix 4.
mandatory parameter

Output Image

The name of a standard TIFF image to be written with a quick look of the specified AOI (the extension "*.tif*" is automatically added by the system).
Example: `Output Image = "preview"`
mandatory OUTPUT
BEST extension: ".tif"

Coordinates Retrieving by Example Image

Description

If a region has been cropped from a quick look image using a non-Toolbox TIFF image processing tool, the COORDINATES RETRIEVING BY EXAMPLE IMAGE function will determine the coordinates that define the cropped region within the original image.

The Coordinates Retrieving function compares two images: an original quick look and a rectangular portion of it (the example image), cropped using an external TIFF image processing tool. The system then returns the coordinates of two opposite corners of the example image, expressed in the full resolution row, column coordinate system of the original image.

This function is useful when the user wants to visually select an AOI using the quick look image in an external TIFF image processor, without considering quantification. By this method, the coordinates of the AOI, necessary for the FULL RESOLUTION EXTRACTION function are easily obtained.

The quick look versions with or without a superimposed grid can both be used but, of course, an original quick look with a grid cannot be compared with an example image without a grid or vice versa.

Some care must be taken with external TIFF image processing freeware used for cropping due to the presence of bugs and malfunctions. For example, the XV tool (version 3.1.0) for Solaris2™ has some problems when cropping a very small image: if the number of columns of the cropped image is less than 72, an error occurs.

When an incorrect example image is input to the COORDINATES RETRIEVING BY EXAMPLE IMAGE function, a warning message is issued explaining that it will not be possible to retrieve the full resolution coordinates. In such cases, try another image processing system.

Typical Processing Chain

HEADER ANALYSIS ⇒ QUICK LOOK GENERATION ⇒ *cropping using external tool* ⇒ COORDINATES RETRIEVING BY EXAMPLE IMAGE

Example "INI" file

```
[COORDINATES RETRIEVING]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "quicklook.tif"
Cropped Tiff Image = "example.tif"
Output Coordinates File = "coords"
```

Parameter Summary: Coordinates Retrieving by Example Image

Input Image

The original quick look image (with or without grid) in standard TIFF format.
Example: `Input Image = "quick look.tif"`
mandatory INPUT

BEST extension: ".tif"

Cropped Tiff Image

An example image cropped from the original quick look image, in standard TIFF format
Example: `Cropped Tiff Image = "example.tif"`
mandatory INPUT
BEST extension: ".tif"

Output Coordinates File

The name of the output text file that will be written with the row, column coordinates of the Top Right and Bottom Left corners of the example image, expressed in the full resolution coordinate system (an extension "*.txt*" is automatically added by the system).
Example: `Output Coordinates File = "coords"`
mandatory OUTPUT
BEST extension: ".txt"

Support Data Ingestion

Description

The SUPPORT DATA INGESTION function converts auxiliary data (e.g. antenna pattern information or lookup tables for calibration) from an ESA ASCII format into the Toolbox internal format.

This operation is only needed if a change to this data occurs and the auxiliary files included in the Toolbox need to be replaced.

Of course, the user is free to ingest his own antenna patterns or ADC lookup tables.

Example "INI" files

The following four "*.ini*" files show how to transform the two antenna patterns and the two ADC lookup tables from the ESA format (an ASCII file with two columns) into the internal file format (note that these files shall be kept in the './cfg' directory).

```
[SUPPORT DATA]
Input Dir = "C:\best\cfg\"
Output Dir = "C:\best\cfg\"
Input Support Data File = "apers1.dat"
Output Image = "apers1"

[SUPPORT DATA]
Input Dir = "C:\best\cfg\"
Output Dir = "C:\best\cfg\"
Input Support Data File = "apers2.dat"
Output Image = "apers2"

[SUPPORT DATA]
Input Dir = "C:\best\cfg\"
Output Dir = "C:\best\cfg\"
Input Support Data File = "adcers1.dat"
Output Image = "adcers1"

[SUPPORT DATA]
Input Dir = "C:\best\cfg\"
Output Dir = "C:\best\cfg\"
Input Support Data File = "adcers2.dat"
Output Image = "adcers2"
```

Parameter Summary: Support Data Ingestion

Input Support Data File

The external file in ASCII format.

Example: `Input Support Data File = "ers1_antpat.dat"`

mandatory INPUT

BEST extension: (ascii input file)

Output Image

The name of the translated file to be written in the Toolbox internal format (an extension "*.SDf*" is automatically added by the system).

Example: `Output Image = "ers1_antpat"`
mandatory OUTPUT
BEST extension: ".SDf"

Import GeoTIFF

Description

The IMPORT GEOTIFF tool converts a GeoTIFF image including its associated annotation data into the BEST internal format.

Important: The following functions cannot be applied to data converted using the IMPORT GEOTIFF tool: OVERSAMPLING, CO-REGISTRATION, SPECKLE FILTER, the Calibration tools and the Data Conversion tool (except GEOMETRIC CONVERSION [(lat, lon) ? (row, col)] and ANCILLARY DATA DUMP).

No AOI is permitted in this operation.

Example "INI" file

```
[IMPORT GEO-TIFF]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "mr_gtif.tif"
Output Image = "int_gtif"
Delete Input Image = "N"
```

Parameter Summary: Import GeoTIFF

Input Image

The external GeoTIFF image.
Example: `Input Image = "mr_gtif.tif"`
mandatory INPUT
BEST extension: ".tif"

Output Image

The name of the output internal format file that contains the input image and annotations.
Example: `Output Image = "int_gtif"`
mandatory OUTPUT
BEST extension: ".GT?" where "?" indicates that the output image retains the same format as the input image.

Import TIFF

Description

The IMPORT TIFF tool converts an image in standard TIFF format to the Toolbox internal format. Any annotations, written in a separate text file, are inserted into the output internal format image. The data to be converted can be initially present on the hard disk or another media, thus avoiding the need to dump the image using the operating system commands.

Example "INI" file

```
[IMPORT TIFF]
Annot Input Dir= "C:\BEST_out\"
Input Annotation = "anno_tif.txt"
Input Dir = "C:\BEST_out\"
Input Image = "ext_tif.tif"
Output Dir = "C:\BEST_out\"
Output File = "imp_tif"
Delete Input Image = "N"
```

Parameter Summary: Import TIFF

Annot Input Dir

The path to the directory that contains the annotation file, if one exists.
Example: `Annot Input Dir = "./"`
mandatory INPUT

Input Annotation

The name of the text file that contains any annotation to be inserted into the output internal format image.
Example: `Input Annotation = "anno_tif.txt"`
mandatory parameter

Input Image

The external image in a standard TIFF format.
Example: `Input Image ="ext_tif.tif"`
mandatory INPUT
BEST extension: ".tif"

Output File

The name of the output internal format file that contains the input image and annotations.
Example: `Output File = "imp_tif"`
mandatory OUTPUT
BEST extension: ".IT?" where "?" indicates that the output image retains the same format as the input image.

Import Raster Image

Description

Using the IMPORT RASTER IMAGE function, it is possible to convert external images not in the CEOS or MPHSPH format (but having similar pixel size) to the BEST internal file format.

Due to the fact that the function operates on pure image data, no annotation is inserted into the output internal format image. Therefore, the number of BEST functions which can process the output from the IMPORT RASTER IMAGE function is limited.

Often the external images will include both a file header section (once for the image) and a line header (for each line). The raster import function is able to skip both header sorts to extract an output dataset containing only the image pixels (instead of a mixture of pixels and header bytes).

Even if no direct AOI can be used, it is possible to define a rectangular AOI using the following parameters:

- File Header Bytes
- Line Header Bytes
- Number of Rows
- Number of Columns

This function, in allowing direct access to the media, can be easily used to extract images with a corrupted or missing header.

Example "INI" file

The following "*.ini*" file is an example for a real raster image conversion (the parameters are those used to convert a 500 rows by 500 columns portion of a CEOS PRI image file from an Exabyte tape unit on a Unix machine):

```
[IMPORT RASTER]
Input Dir = "./"
Output Dir = "./"
Input Media Type = "tape"
Input Image = "/dev/rst1"
Media File Skip = 2
Data Type = "2I"
File Header Bytes = 16012
Line Header Bytes = 12
Image Record Length = 16012
Number of Rows = 500
Number of Columns = 500
Swap Bytes = "N"
Output Image = "imported_img"
```

Parameter Summary: Raster Image Import

Input Media Type

The type of media on which the raster image is held, chosen between:

- "disk" (hard disk)
- "tape" (Exabyte)

- "cdrom"

Example: `Input Media Type = "disk"`
optional parameter (default is "disk")

Input Image

When 'Input Media Type' is set to "disk" or "cdrom", this parameter gives the name of a 2I or complex 2I image in RASTER format; when 'Input Media Type' is set to "tape" it gives the device name of the tape unit:

- for an image held on the hard disk use:
  `Input Image = "external_img.dat"`
- for an image held on an Exabyte tape use:
  `Input Image = "/dev/rst1"`

mandatory INPUT
BEST extension: (data product)

Data Type

The type of RASTER data to be imported:

- "2I" (16-bit real image)
- "Complex 2I" (complex image, 16 bits + 16 bits)

Example: `Data Type = "Complex 2I"`
mandatory parameter

Media File Skip

The number of files that precede the image data file to be imported; these files will be skipped. This parameter is not used when 'Input Media Type' is set to "disk" or "cdrom".
Example: `Media File Skip = 2`
optional parameter (default is "0")

File Header Bytes

The number of bytes to skip once at the beginning of the image data file; typically these bytes constitute the file header section before the image data itself.
Example: `File Header Bytes = 16012`
optional parameter (default is "0")

Line Header Bytes

The number of bytes to skip at the beginning of each image line; typically these bytes constitute the header section of each line and contain non-image data.
Example: `Line Header Bytes = 12`
optional parameter (default is "0")

Image Record Length

The length of the image data file, expressed as number of bytes.
Example: `Image Record Length = 16012`
mandatory parameter

Number of Rows

The number of rows of the input image to be imported.
Example: `Number of Rows = 500`
mandatory parameter IF 'Input Media Type' is set to "tape"
optional parameter in the remaining cases (default is entire image)

Number of Columns

The number of columns of the input image to be imported.
Example: `Number of Columns = 500`
optional parameter (default is entire image)

Swap Bytes

A flag indicating whether the order of each byte couple shall be swapped before writing in the output file. Use "Y" to execute the swapping when reading a CEOS product (which is stored in a NONDEC format) with a PC; set to "N" to leave the byte ordering untouched when reading a MPHSPH product (which is stored in DEC format) with a PC (PCs are DEC ordering machines).
Example: `Swap Bytes = "Y"`
optional parameter (default is "N")

Output Image

The name of the file to be written in the Toolbox internal format.
Example: `Output Image = "imported_img"`
mandatory OUTPUT
BEST extension: “.RIs” for real data; “.RIt” for complex data

New Product Adding

This is not really a function in the same sense as the other SAR Toolbox algorithms but instead makes it possible to recognise and decode SAR products that were not previously recognised as standard products.

The New Product Adding function permits the insertion of new format in the set of those recognized by STB. The new format descriptor files shall all be kept in the directory CFG.

To make possible the decoding of a new product, the following operations are required:

generation of a new section in the file in which all the nominal values of the media structure data are kept
generation of the FDF file which contains the detailed format structure (following the FDF syntax explained in the next section)
storing of the FDF file in the configuration directory with a proper naming convention

After these operations the system is capable to accept this new product and treat as a standard products. In the next paragraphs these operations are described in detail.

Nominal Media Structure File description

As already described in the previous paragraph related to the Media Analysis, the value of the following parameters:

the product format
the product type
the sensor Id
the source PAF

is obtained by a comparison of the media structure (obtained scanning the media, CDRM or EXABYTE) with a file containing the nominal values of this structure for the various products. This file has then as many entries as the number of products which are different each other, in their media structure. Obviously, if two product have the same media structure, there is no way to guess the format, type, sensor and PAF. In the next page is showed a listing showing the nominal media structure file.

A typical entry is the Nominal Media Structure File has the following structure:
PRODUCT PARAMETERS SECTION
VOLUME SECTION *1*
VOLUME SECTION *2*
.....
VOLUME SECTION *i*
.....
VOLUME SECTION *N*

Within the PRODUCT PARAMETERS SECTION
the structure is:
[*product_type*]
SENSOR: *sensor_id*
FORMAT: *format_id*
SOURCE: *source_id*
where:

product_type can be one of the following three character strings (SLI stands for SLCI product):

RAW
SLC
SLI
PRI
GEC
GTC

sensor_id can be ERS1 or ERS2 or in general a four character string XXXN; the various sensors are distinguished only by the last character N of their conventional name (so avoid using ERS1 and ENV1 as two different sensors)

format_id can be CEOS or MPHSPH or in general a n-character string FXX....X, the various formats are distinguished only by the first character F (so avoid using CEOS and CGM as two different formats)

source_id can be:

ESP (for ESRIN products)
DEP (for DPAF products)
UKP (for UKPAF products)
ITP (for IPAF products)
SIS (for SINGAPORE products)

or in general a three character string; all the three character are used to identify the source processing facility

Within a VOLUME SECTION *i*
the structure is:
VOLUME: *i*
FILE SECTION *1*
FILE SECTION *2*
.....
FILE SECTION *j*
.....
FILE SECTION *M*
where:

i is a integer number (which increases with each volume)

Within a FILE SECTION *j*
the structure is:
FILE: *j* *comparation_flag*
RECORD SECTION *1*
RECORD SECTION *2*
.....
RECORD SECTION *k*
.....
RECORD SECTION *S*
where:

j is a integer number (which increases with each file)

comparation_flag which can be *Y* or *N*, tell to the system to use or not this information during the comparation of the media structure obtained scanning the media with the values in the nominal media structure file

Each RECORD SECTION *k*
has the following aspect:
RECORD: *recs_number* *rec_lencomparation_flag*
where:

recs_number is the number of sequential records of the same size

rec_len is the record length in bytes, common to all the *rec_len* records

comparation_flag which can be *Y* or *N*, tell to the system to use or not this information during the comparation of the media structure obtained scanning the media with the values in the nominal media structure file

An example of one entry of the nominal media structure file is:

```
[PRI]
SENSOR: ERS1
FORMAT: CEOS
SOURCE: ITP
VOLUME: 1
FILE: 1 Y
RECORD:    4   360 Y
FILE: 2 Y
RECORD:    1   720 Y
RECORD:    1  1886 Y
RECORD:    1  1620 Y
RECORD:    1  1046 Y
RECORD:    1 12288 Y
FILE: 3 Y
RECORD:  8201 16012 Y
FILE: 4 Y
RECORD:    1   360 Y
```

This entry describes a ERS1 PRI product of the Italian PAF

kept on 1 Volume

organized on 4 files

file 1 has 4 records of 360 bytes each

file 2 has

1 record of 720 bytes

1 record of 1886 bytes
1 record of 1620 bytes
1 record of 1046 bytes
1 record of 12288 bytes
file 3 has 8201 records of 16012 bytes each
file 4 has 1 records of 360 bytes

All this information is used during the comparison of the media structure (all flags are Y). Following these rules, it is possible to add new entries in the Nominal Media Structure File.

Format Description File description

In order to have a tool independent from the product type (PRI, SLC, GEC, etc.) and media (EXABYTE or CDROM) to be de-formatted, a simple description language has been defined to describe the format of each product in a "*Format Description File*" (FDF). In this way, a change of product format does not affect the source code of any module: it is only necessary to properly update the FDF of the correspondent product type. Note that this approach allows to handle in the same manner not only the CEOS format, but also all the positional product-formats (for example, the MPH-SPH product format).
Each *Product Annotation* must be given an annotation name: this name is used to find out the tags related to the annotation itself when inserting it into the internal format image. The format-specifications characteristics of annotations are meaningful to properly compute the size of buffers to be used when reading the product format. These information are all included into the FDF file.
It is allowed the use of variables. Two kinds of variables are available:

internal variables (named *global variables*);
external variables.

The first ones are useful for two main purposes:

to store some values that could be necessary later (in future steps, that could come when that values are no more available if they are not stored in temporary and globally accessible variables);

to access values that are automatically maintained and updated by the de-formatter itself (such as the current date, the number of bytes read from the current record, the current number of processed records, and so on).

There exist a set of pre-defined global-variables the programmer can use. In addition, he can define his own set of global variables and can assign values to them. If useful, they can also be initialized; for this purpose, an apposite optional section of the FDF (named *declaration section*) is provided.

The following sections can be found into a description file:
Declaration Section
It contains the declaration and the initialization of the global variables that must have an initial value. This section should at least contain the variables which describes the various files and record sizes of the product, now described using as example a CEOS PRI. Note that the symbol "@" is part of the variable name and should not be omitted.

Variable	Example value for a CEOS PRI	Comment
@MAX_F_SZ(1)	360	the maximum record size for the file #1; use 0 for variable length records of a number for a specified record size
@MAX_F_SZ(2)	0	the maximum record size for the file #2; note that the second file corresponds to the Leader file which has variable length records
@MAX_F_SZ(3)	0	the maximum record size for the file #3; note that the third file corresponds to the Imagery file which could have variable length records
@MAX_F_SZ(4)	360	the maximum record size for the file #4
@F_R(1,1)	360	the actual (fixed) record size of the file #1, record #1
@F_R(1,2)	360	the actual (fixed) record size of the file #1, record #2
@F_R(1,3)	360	the actual (fixed) record size of the file #1, record #3
@F_R(1,4)	360	the actual (fixed) record size of the file #1, record #4
@F_R(2,1)	0	the record size of the file #2, record #1, set to variable length
@F_R(2,2)	0	the record size of the file #2, record #2, set to variable length
@F_R(2,3)	0	the record size of the file #2, record #3, set to variable length
@F_R(2,4)	0	the record size of the file #2, record #4, set to variable length
@F_R(2,5)	0	the record size of the file #2, record #5, set to variable length
@F_R(3,1)	0	the record size of the file #3, record #1, set to variable length
@F_R(3,2)	0	the record size of the file #3, record #2, set to variable length
@F_R(4,1)	360	the actual (fixed) record size of the file #4, record #1
@BITS_PER_SAMPLE	16	the size of each image pixels in bits
@SAMPLE_PER_PIX	1	the number of layers in the image; use 1 for real images and 2 for complex images
@IMG_REC_NO	0	initialization of the variable which contain the num ber of records of the image section
@IMG_REC_SZ	0	initialization of the variable which contain the record size in bytes of the image records
@START_LINE	0	initialization of the variable which contain the start line of the image contained in the actual volume
@END_LINE	0	initialization of the variable which contain the end line

		of the image contained in the actual volume

Structure Section

It describes the main structure of the product to which the description file is referred. Since a product generally consists of more than one file, this section describes:

File Number: indicates the file ordinal number within the product;

Product File Name: a self explanation string about the file;

Product File Code: the extension used for the file onto the disk;

Presence-Type Code and *Can-Be-Interrupt* flag: these two fields encode how products can span over more than one medium (when necessary because of the product dimensions compared with medium capacity);

File Format Code: the physical format (fixed length, variable length, etc.)

File Record Length: (optional) the default record-size for each file.

File Sections

Each file-section (one section for each file of the actual product) describes the structure of the file it refers to. Since a file can be considered as a sequence of records, a file section consists of a sequence of record sections, each of one describes the logical format of the referred record.

Record Sections

Since a record can be considered as a sequence of fields, each *record section* is a sequence of *field sections*, each of one describes the logical and physical format of the field which it refers to. Note that a *file section* can group several *record sections* into blocks (using the '*repeat construct*'), when iterations on that set of records are required.

Field Sections

A field-section describes how the related field has been filled (when de-formatting) or must be filled (when formatting). Each field-section is composed by the following sub-sections:

Field number: indicates the field ordinal number within the record;

Field offset: indicates the starting bytes as an offset from the beginning of the record;

Internal format flag: it is a flag (the value can be Y or N) that specifies if the field has to be considered a 'product annotation' (that means, characteristic of the product, independently from its format) or a 'format annotation' (that means, characteristic of the product format but not of the product data itself). Note that all the 'product annotations' are also inserted into the product file in internal format, whereas the 'format annotations' are not included into the product, as they can be obtained from the header listing file, if necessary

Align type: can be L, R or C, indicating if the field-value has to be respectively left- aligned, right-aligned or centered aligned (respect to the field-size). As a default, numeric fields are generally left-aligned whereas text-fields are right-aligned;

Field format: it describes the formatting convention to be used to decode the field. The allowed formats are:

Strings

Annn (where 'nnn' is any integer value), that means a string of nnn characters or even a dummy field of 'nnn' bytes

Integer or floating numbers coded as strings

In, that means an integer field expressed using its ASCII representation, with a fixed number (n) of digits. For example, if the 128 integer-number has to be used to fill a I4 field, the four characters will fill the field: '1', '2', '8' and ' ';

Fnn.mm, representing a real number expressed using its ASCII representation, where 'nn' and 'mm' indicate, respectively, the number of digits before and

after the decimal point (for example: F16.7 means a real number of 16 digits, 7 of which after the decimal point).

Dnn.mm, representing a real number in double-precision, expressed using its ASCII representation, where 'nn' and 'mm' indicate, respectively, the number of digits of the entire number and after the decimal point (for example: D22.10 means a double of 22 digits, 10 of which after the decimal point).

Ennn.mm, representing a real number in exponential format, expressed using its ASCII representation, where 'nn' and 'mm' indicate, respectively, the number of digits (counting also the ‘E’ and the + and - sign) of the entire number and after the decimal point.

Integer numbers in binary representation

Bn, that means a binary field of length n (i.e.: a sequence of n bytes), with n equal to 1, 2 or 4 in NODEC byte ordering.

Kn, that means a binary field of length n (i.e.: a sequence of n bytes), with n equal to 1, 2 or 4 in DEC byte ordering.

Image data

The image data records are recognized by the condition of a REPEAT statement followed by a single record definition. The REPEAT shall use a difference between the two reserved global variables START_LINE and END_LINE. These two variables shall appear defined in the FDF and associated with a couple of fields.

Rn, that means image sample with a pixel having 16 bit unsigned integer format, in NODEC byte ordering. The number ‘n’ is not used in the subsequent processing. If this field starts at a byte position different from 1, the previous bytes are considered as line header and not inserted in the image

Vn, that means image sample with a pixel having 16 bit unsigned integer format, in DEC byte ordering. The number ‘n’ is not used in the subsequent processing. If this field starts at a byte position different from 1, the previous bytes are considered as line header and not inserted in the image

Cn, that means image sample with a pixel having 8 bit signed integer format. The number ‘n’ is not used in the subsequent processing. If this field starts at a byte position different from 1, the previous bytes are considered as line header and not inserted in the image

Tag name: it is the symbolic name of the field; the system uses this name to retrieve the associated tag (a 16 bits number greater equal than 35001 which is used in the internal format image as field identificator); if a new tag has to be defined, it has to be inserted in the include file TAGS.H (kept in the configuration directory CFG), which contains all the correspondences between symbolic names and tag numbers; some care must be taken in the usage of names and tag numbers not already defined

Future Extension #1: reserved for future extensions

Global Variable Assignment: used to contain a reference to a global-variable, that will be assigned with the result of the evaluation of the field-value section; it is used to store temporary values that must be used during subsequent de-formatting steps (for example, the number of records contained in a subsequent section, such as the number of state-vectors, number of image lines and so on)

Future Extension #2: reserved for future extensions

Future Extension #3: reserved for future extensions

Field Name: contains the detailed name of the field (as appears in the header listing file obtained in the header decoding function)

Units: contains the measure units of the field (as appears in the header listing file obtained in the header decoding function)

Comment: contains a comment about the field (as appears in the header listing file obtained in the header decoding function)

Naming conventions of the FDF files

The name convention used to identify the various FDF files is mainly based on the restriction to 8 character imposed by the need that the system can run on DOS, which cannot handle long names. The information about the product type, the sensor, the format, the source facility and also the single or multi volume are coded in the following way.

The FDF name is PTYSFSRC.SMV where:

PTY is the three character product type (RAW, SLC, SLI, PRI, GEC, GTC)

S is the single digit number representing the sensor id (1,2)

F is the single character format id (C, M)

SRC is the three character source station (ITP, DEP, UKP, DEP, SIS or a new station)

SMV which is the extension, can be

CFS for single volume products

CFF for multi volume products, first volume

CFC for multi volume products, center volume(s)

CFL for multi volume products, last volume

The various sections of the DFD name shall follow the same values used in the Nominal Media Structure File. As an example, if in the Nominal Media Structure File there is an entry describing a PRI product, coming from a certain ACM station, in a format CEOS, acquired by a ERS2 sensor, split on 1 volume, the corresponding FDF file shall be named: PRI2CACM.CFS.

If was split on two volumes the two FDF shall be named PRI2CACM.CFF and PRI2CACM.CFL, while if was split on three volumes the three FDFs should be PRI2CACM.CFF, PRI2CACM.CFC, PRI2CACM.CFL.

8. Data Export

This chapter documents the following tools:

1. Export GeoTIFF
Converts data from internal format to a GeoTIFF image that includes geographic information.

2. Export to TIFF
Converts 8 bit data from the Toolbox internal format to standard TIFF format as either single-channel greyscale or 3-channel colour images.

3. Export to BIL
Converts one or more (up to ten) internal Toolbox format images having the same size and data type to one binary image in BIL (Band Interleaved by Line) format.

4. Export to RGB
Converts three internal Toolbox format images with the same size to a 24-bit RGB image.

Export GeoTIFF

Description

The EXPORT GEOTIFF tool is based on the functions of the related handling library.

The GeoTIFF format is a variation of the TIFF image file format which additionally conveys geographic information.

No AOI is permitted in this export operation.

Example "INI" file

```
[GEO-TIFF GENERATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "asar_apm.XTs"
Output Image = "exp_gtif"
Delete Input Image = "N"
```

Parameter Summary: Export GeoTIFF

Input Image

The image to be exported to the GeoTIFF format.
Example: `Input Image = "asar_apm.XTs"`
mandatory INPUT
BEST extension: ".??i", ".??s", ".??t", ".??f" or ".??c" where "??" indicates that any BEST module could have produced the file.

Output Image

The name of the output GeoTIFF file containing the image and geographic annotations
Example: `Output Image = "exp_gtif"`
mandatory OUTPUT
BEST extension: ".tif"

Export to TIFF

Description

The EXPORT TO TIFF function converts an image in the internal BEST format to a universally readable TIFF format.

A standard grey-level TIFF image can be generated from data of any type handled internally by the Toolbox, i.e. 8-bit integer, 16-bit integer, floating point or complex pixels. An RGB colour TIFF image can be generated from three 8-bit images.

The TIFF version for such export is TIFF6. An ASCII file containing the image annotations is also generated as an output.

No AOI is permitted in this operation.

Important: If the image viewer XV is used to visualise the output from the TIFF conversion module, it may be necessary to first launch the XV software and then load the image using the internal commands, rather than using, for example, the command:

```
xv grey_img.tif
```

This is because of the nature of the TIFF file generated by the Toolbox module.

Example "INI" files

The following "*.ini*" file is an example for grey-level TIFF image generation:

```
[TIFF GENERATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Images = "asar_apm.XTs"
Delete Input Image = "N"
Output Image = "apm_tif"
Output Annotations File = "anno_tif"
```

The following "*.ini*" file is an example for 3-colour TIFF image generation:

```
[TIFF GENERATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Images = "red.GCi","green.GCi","blue.GCi"
Delete Input Image = "N"
Output Image = "rgb_img"
Output Annotations File = "anno_rgb"
```

Parameter Summary: Export to TIFF

Input Images

The name of internal format image(s) to be converted. Where three images are listed (for an RGB TIFF), they should be in the order "red channel, green channel, blue channel".
Example: `Input Image = "asar_apm.XTs"`
mandatory INPUT

BEST extension: “.??i”, “.??s”, “.??t” , “.??f” or “.??c” where “??” indicates that any BEST module could have produced these files.

Output Image

The name of the output standard TIFF image (the extension “.tif” is automatically added by the system).
Example: `Output Image = "apm_tif"`
mandatory OUTPUT
BEST extension: “.tif”

Output Annotations File

The name of the output ASCII file containing the annotation data (the extension “.txt” is automatically added by the system).
Example: `Output Annotations File = "anno_tif"`
mandatory OUTPUT
BEST extension: txt

Export to BIL

Description

The EXPORT TO BIL tool converts one or more (up to ten) real or complex images into a binary file arranged in the 'band interleaved by line' (BIL) format. A maximum of 10 images in the BEST internal format can be submitted as inputs as long as they all share the same data type (integer or floating point) and size. The output can be read by many image processing software packages. The process maintains the pixel format, and therefore the accuracy of the source data.

The conversion generates an output image file (with the extension "*.BG*"), an associated ASCII header file (with the extension "*.ers*") and a text file containing the annotations of the first input image (with the extension "*.txt*"). The "*.ers*" file is not generated if the inputs are complex images.

No AOI is permitted in this conversion.

For a data set of *z* bands with dimensions *y* rows and *x* columns, the data in the binary file will be arranged as follows:

```
(band 1 row 1 pixel 1)......(band 1 row 1 pixel x)
(band 2 row 1 pixel 1)......(band 2 row 1 pixel x)
......
(band z row 1 pixel 1)......(band z row 1 pixel x)

(band 1 row 2 pixel 1)......(band 1 row 2 pixel x)
(band 2 row 2 pixel 1)......(band 2 row 2 pixel x)
......
(band z row 2 pixel 1)......(band z row 2 pixel x)

......

(band 1 row y pixel 1)......(band 1 row y pixel x)
(band 2 row y pixel 1)......(band 2 row y pixel x)
......
(band z row y pixel 1)......(band z row y pixel x)
```

Example "INI" files

```
[BIL GENERATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Images = "input.XTs"
Output Image = "output"
Output Annotations File = "output_annot"
```

The following example uses 4 input files:

```
[BIL GENERATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Images = "band1.XTs","band2.XTs","band3.XTs","band4.XTs"
Output Image = "bil_img"
Output Annotations File = " band1_annot"
```

Parameter Summary: Export to BIL

Input Images

The input image list.

Example: `Input Images = "band1.XTs","band2.XTs","band 3.XTs","band4.XTs"`

mandatory INPUT

BEST extension: “.??i”, “.??f”, “.??c”, “.??s” or “.??t” where "??" indicates that it is not important which module created the files, as long as the data type is correct.

Output Image

The name of the BIL output image containing multi-band data (an extension “*.BG*” is added by the system) and the associated ASCII header file (an extension “*.ers*” is added by the system).

Example: `Output Image = "tiff_img"`

mandatory OUTPUT

BEST extension: “.BG” and “.ers”

Output Annotations File

The name of the output file containing the annotations data of the first listed input image (an extension “*.txt*” is added by the system)

Example: `Output Annotations File = "output_annot"`

mandatory OUTPUT

BEST extension: “.txt”

Export to RGB

Description

The EXPORT TO RGB tool converts three internal Toolbox format images with the same size into a 24-bit RGB image, which can be read by other image handling software packages. Only images with a single sample per pixel can be given as input.

No AOI is permitted with this tool.

Example "INI" file

```
[RGB GENERATION]
Input Dir = "C:\BEST\rgb\"
Output Dir = "C:\BEST\rgb\"
RGB Images = "ima1.XTs", "ima2.XTs", "ima3.XTs"
Output Image = "rgb"
```

Parameter Summary: Export to RGB

RGB Images

The names of the three internal format images to be written to the channels of the RGB file. They should be entered in the order "red channel, green channel, blue channel".
Example: `RGB Images = "ima1.XTs", "ima2.XTs", "ima3.XTs"`
Mandatory INPUT
BEST extension: ".??i", ".??s", ".??f" where "??" indicates that it is not important which module created the files, as long as the data type is correct.

Output Image

The name of the output RGB image. The extension "*.tif*" is automatically added by the system.
Example: *`Output Image = "rgb"`*
Mandatory OUTPUT
BEST extension: ".tif"

9. Data Conversion

This chapter documents the following tools:

1. Gain Conversion
Rescales floating-point or real 16-bit integer data to 8 bits, thereby preparing it for export to formats that can be visualised in basic graphics packages.

2. Power to Amplitude Conversion
Converts a power image into an amplitude image.

3. Amplitude to Power Conversion
Converts an amplitude image into a power image.

4. Linear to dB Conversion
Converts an amplitude or intensity image with a linear scale into an image in decibel (dB) units.

5. Complex to Amplitude Conversion
Derives the amplitude modulus from a complex image.

6. Integer to Float Conversion
Converts a real image from the integer format to the floating-point format.

7. Ancillary Data Dump
Generates an ASCII listing of the image annotations relating to an image in the Toolbox internal format.

8. Image Operation
Performs basic algebraic operations (sum, subtract, multiply or divide) between two images or between one image and a constant factor. It is also possible to calculate the absolute value of a single image.

9. Geometric Conversion
Converts between row, column and latitude, longitude coordinates for points specified in any given image. Also calculates the satellite's position and angles of incidence and look for the specified points.

10. Slant Range to Ground Range Conversion
Reprojects images from slant range (range spacing proportional to echo delay) to ground range (range spacing proportional to distance from nadir along a predetermined ellipsoid). The tool works on complex data (extracted and/or co-registered SLC products) and real data (coherence products).

11. Flip Image
Executes a horizontal or vertical flip operation (or both) on any internal Toolbox format image.

12. Sensitivity Vector Evaluation
Calculates the sensitivity vector of an input image point by point.

Gain Conversion

Description

The GAIN CONVERSION function operates on a floating point or 16-bit integer real image, converting it to an 8-bit image. This tool will often be used to convert the pixel values in an image into a range suitable for visualisation. Therefore, the tool is often applied prior to exporting to the TIFF format.

In a SAR image, 99% of the image data may have pixel values between 0 and 1000. However, the maximum value of the image may be as great as 30,000. If the conversion from 16 bit to 8 bit is done with a simple scaling, the resultant image will appear all black, except for a very few isolated pixels having very high values. The GAIN CONVERSION tool has facilities to allow the conversion to be made in such a way that the image can be more sensibly visualised.

The only Area of Interest that is permitted is the rectangular AOI with corners expressed in the row, col system.

The gain conversion module allows the following three modes of operation. The mode is selected by a parameter specified in the "*.ini*" file.

(i) Fixed Gain Conversion. In this mode all of the input image pixel values are divided by a fixed gain value, selected by the user. If the pixel values, after the division, exceed 255 they are saturated at 255. In this way a very simple radiometric stretch scheme can be implemented. The limitation of this method is that the optimum scaling will be dependent on the kind of scene contained in the SAR product. The gain constant will therefore need to be chosen carefully or by using a process of trial and error.

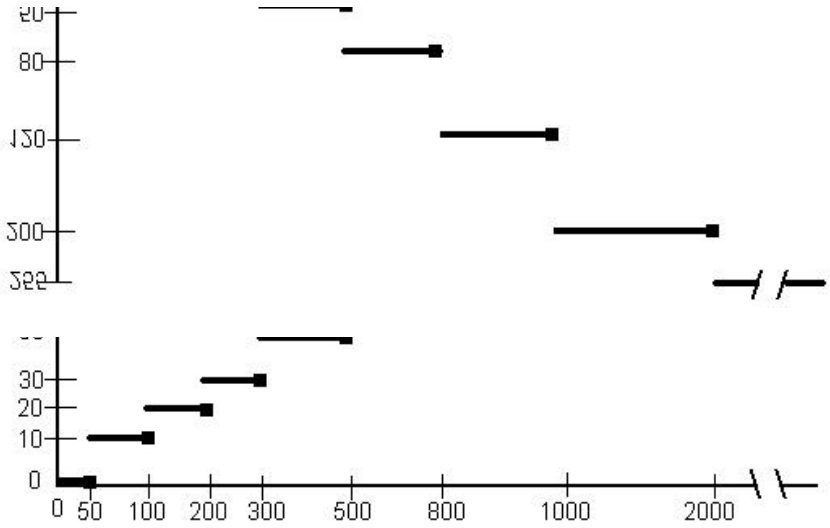
(ii) Variable Gain Conversion. In this mode a linear stretch is performed on those pixels with values that fall between two upper and lower radiometric values, (k_b and k_a respectively). The values k_b and k_a are obtained from the histogram of the image. The user is required to select two percentage values, Min Percentage and Max Percentage (for example, 1% and 99.5%). The two radiometric values k_a and k_b are then calculated, such that Min Percentage (e.g. 1%) of the image pixels have values below k_a and Max Percentage (e.g 99.5%) of the image data have pixel values below k_b. Pixels with values falling between the limits k_a and k_b have a linear stretch applied to them. Pixels with values outside of this range are saturated at 0 and 255.

An optional parameter (Number of Black Levels) can be used to exclude pixels with low values from the calculation, which is used to evaluate the two radiometric values k_b and k_a. If, for example, Number of Black Levels is set to 2.0, then all of those pixels with values lower than 2.0 are not included in the histogram on which the Min Percentage and Max Percentage levels are drawn. The Number of Black Levels parameter allows the computation of a meaningful statistics even for images containing a large portion of pixels near or equal to zero. In particular this is useful for obtaining a good image appearance for GEC or GTC products which contain large regions of black pixels due to the rotation that is applied to these images.

(iii) Look-Up Table Mode. In this mode the user is explicitly required to give the piecewise function that is used to stretch the image. This lookup table should be in the form of an ASCII file contain pairs of numbers which define the way in which the values of the input image are mapped onto the 256 intensity values that are available for the 8 bit output image.

The following examples illustrate the format that is used in the look-up tables.

Look-Up Table Format for Gain Conversion, Example 1



In the figure, above, the x-axis represents the input values and the y-axis represents the output values. The mapping between this input and output is described by the following look-up table:

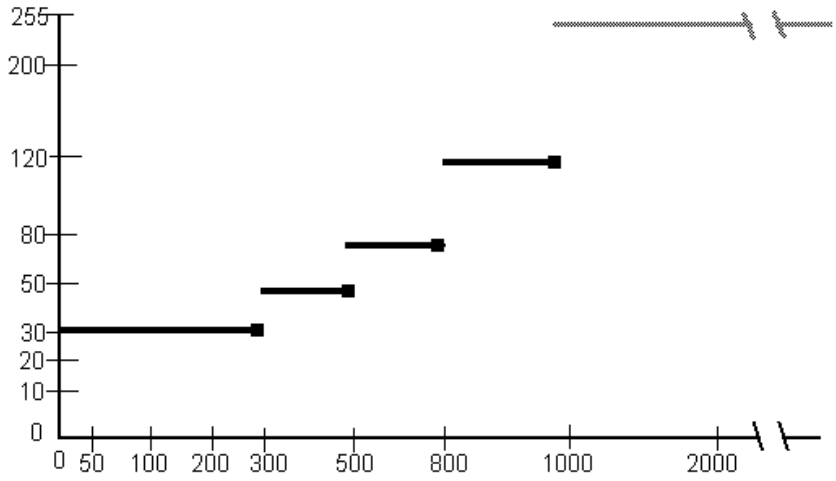
0	50
10	100
20	200
30	300
50	500
80	800
120	1000
200	2000

The first pair of numbers are 0 and 50, this indicates that all pixels in the input image with a value 50 or less are mapped to 0 in the output image. The second pair of numbers are 10 and 100, showing that all pixels in the input image that are between 51 and 100 are mapped to 10 in the output image. The third pair is 20 and 200, indicating that the input pixels with values in the range 101 to 200 are mapped to 20 in the output image.

As can be seen from this example it is not necessary for there to be a separate pair of entries for all of the possible 256 output values.

Also it should be noted that it is not required that the first output value refers to 0 and the last to 255 If the entry corresponding to the output value 255 is missing (as in the example above), the algorithm assumes that all the input values greater than the last present in the lookup shall be set to 255. So, in the example above, all pixels in the input image with values greater than 2000 are set to 200.

Look-Up Table Format for Gain Conversion, Example 2



The lookup corresponding to this figure above is:

30	300
50	500
80	800
120	1000

Notice how the input values greater than 1000 are automatically set to 255.

Typical Processing Chain

[HEADER ANALYSIS] -> [FULL RESOLUTION]->[GAIN CONVERSION]->[TIFF GENERATION]

Example "INI" files

The following “.INI” file is an example for the gain conversion with fixed gain.

```
[GAIN CONVERSION]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
Output Image = "fixgain"
```

The following “.INI” file is an example for the gain conversion with variable gain.

```
[GAIN CONVERSION]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
Min Percentage = 0.1
Max Percentage = 99.8
Number of Black Levels = 1.0
Output Image = "vargain"
```

The following “.INI” file is an example for the gain conversion with lookup table.

```
[GAIN CONVERSION]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
User LUT = "lut.dat"
Output Image = "lutgain"
```

Gain Conversion Summary Table

Parameter	Description Example	Comment
Input Image	the name of the real image in internal format Example: Input Image = "t1_priimage.XTs"	mandatory INPUT BEST extension: ??i, ??f, ??s where the "??" indicates it is not important which module created the files, as long as the data type is correct.
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
Scaling Factor	the constant used to scale the input pixel val ues Scaling Factor = 4.0	mandatory parameter for the fixed gain conversion
Min Percentage	the percentage of data at low pixel values which shall be saturated to 0 in the output image, excluding in this count the data hav ing a pixel value less or equal to Number of Black Levels Min Percentage = 0.1	mandatory parameter for the variable gain stretch ing mode
Max Percentage	the percentage of data which will be scaled linearly, excluding in this count the data hav ing a pixel value less or equal to Number of Black Levels and also the data sat urated to 0 by the parameter Min Per centage Max Percentage = 99.8	mandatory parameter for the variable gain stretch ing mode
Number of Black Levels	starting level of the “valid” image pixels; values below this one, are not considered during the histogram evaluation Number of Black Levels = 1.0	optional parameter for the variable gain stretching mode (shall be avoided for the remaining conver sions)
User LUT	the name of the ASCII file containing the lookup table used to stretch the image (in the lookup mode) User LUT = "lut.dat"	mandatory parameter for the user lookup table stretching mode
Output Image	the name of the output image in internal format containing the image converted in 8 bit pixel modulus data (an extension “GCi” is automatically added by the system) Example: Output Image = "cnvt_image"	mandatory OUTPUT BEST extension: GCi

Power to Amplitude Conversion

Description

The POWER TO AMPLITUDE CONVERSION tool takes the square root of the input image pixel values, thus generating a floating-point image representing the amplitude of a power image.

In the output image annotation, the pixel type is set to “amplitude”, so that those tools that need amplitude data as input data can first execute a check.

The only AOI permitted is the rectangular AOI with corners expressed in (row, col).

Example "INI" file

```
[POWER TO AMPLITUDE]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "power_data.APf"
Output Image = "ampl_data"
```

Parameter Summary: Power to Amplitude Conversion

Input Image

The name of the input real image in internal format.
Example: `Input Image = "power_data.APf"`
mandatory INPUT
BEST extension: “.??f” where “??” indicates that any BEST module could have produced this file

AOI specification

see Appendix 4; for polygonal AOI the surrounding rectangular AOI is used
optional parameter (default is entire input image)

Output Image

The name of the output image containing amplitude data (the extension “*.PAf*” is automatically added by the system)
Example: `Output Image = "ampl_data"`
mandatory OUTPUT
BEST extension: “.PAf”

Amplitude to Power Conversion

Description

The AMPLITUDE TO POWER CONVERSION tool computes the square of the input image pixel values, thus generating a floating-point image representing the power of an amplitude image.

In the output image annotation, the pixel type is set to “power”, so that those tools that need power data as input data can first execute a check.

The only AOI permitted is the rectangular AOI with corners expressed in (row, col).

The tool works both with real images and complex data; in the latter case, the square modulus is computed as output. This feature can replace the use of the pipeline between the modulus extraction (COMPLEX TO AMPLITUDE CONVERSION) and the AMPLITUDE TO POWER CONVERSION tools necessary for most complex data processing (see below).

Typical Processing Chain

HEADER ANALYSIS ⇒ FULL EXTRACTION ⇒ AMPLITUDE TO POWER CONVERSION

Example "INI" file

```
[AMPLITUDE TO POWER]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "ampl_data.PAf"
Output Image = "power_data"
```

Paremeter Summary: Amplitude to Power Conversion

Input Image

The name of the input real image in internal format
Example: `Input Image = "ampl_data.PAf"`
mandatory INPUT
BEST extension: “.??f”, “.??i”, “.??s” where “??” indicates that any BEST module could have produced this file

AOI specification

see Appendix 4; for polygonal AOI the surrounding rectangular AOI is used
optional parameter (default is entire input image)

Output Image

The name of the output image containing power data (the extension “*.APf*” is automatically added by the system)
Example: `Output Image = "power_data"`
mandatory OUTPUT
BEST extension: “.APf”

Linear to dB Conversion

Description

The linear to dB conversion function is used to convert an image (an amplitude or an intensity image) from the linear scale to a dB scale.

The AOI which are permitted are the rectangular AOI (with corners expressed in row, col system or in lat, lon) or even the polygonal AOI (in this case the surrounding rectangular AOI is used). No further parameter is needed. Note that to convert a complex image into dB, a modulus extraction shall be executed.

Example "INI" file

[LINEAR TO DB]
Input Dir = "./"
Output Dir = "./"
Input Image = "pwdata.APf"
Output Image = "data_db"

Linear to dB Conversion Summary Table

Parameter	Description Example	Comment
Input Image	the name of the amplitude or power image in internal format Example: Input Image = "data.PAf" or Input Image = "data.APf"	mandatory INPUT BEST extension: ??f, ??i, ??s where the "??" indicates it is not important which module created the files, as long as the data type is correct.
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
Output Image	the name of the output image in dB units (an extension “DBf” is automatically added by the system) Example: Output Image = "data_db"	mandatory OUTPUT BEST extension: DBf

Complex to Amplitude Conversion

Description

The complex to amplitude function works on complex images extracting the modulus, generating a floating point image containing hence amplitude data.

The associated output image annotations which define the pixel type are set to “amplitude”, so that those tools which need amplitude data as input data can first execute a check.

The only AOI which is permitted is the rectangular AOI with corners expressed in row, col system.

Example "INI" file

[COMPLEX TO AMPLITUDE]
Input Dir = "./"
Output Dir = "./"
Input Image = "slc_data.XTt"
Output Image = "modul_data"

Complex to Amplitude Summary Table

Parameter	Description Example	Comment
Input Image	the name of the complex image in internal format Example: Input Image = "slc_data.XTt"	mandatory INPUT BEST extension: ??t, ??c where the "??" indicates it is not important which module created the files, as long as the data type is correct.
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
Output Image	the name of the output image containing the modulus data (an extension “APf” is auto matically added by the system) Example: Output Image = "modul_data"	mandatory OUTPUT BEST extension: CAf

Integer to Float Conversion

Description

The integer to float conversion function works on integer images generating the corresponding floating point image. The only AOI which is permitted is the rectangular AOI with corners expressed in row, col system.

Important: Please note that the title of the function in the "INI" file is [PIXEL TO FLOAT] rather than [INTEGER TO FLOAT]. This inconsistency will be changed in a later version of the SAR Toolbox.

Example "INI" file

[PIXEL TO FLOAT]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Top Left Corner = 0, 0
Bottom Right Corner = 799, 799
Output Image = "float_img"

Integer to Float Summary Table

Parameter	Description Example	Comment
Input Image	the name of the integer image in internal format Example: Input Image = "t1_priimage.XTs"	mandatory INPUT BEST extension: ??i, ??s, ??r, ??t where the "??" indicates it is not important which module created the files, as long as the data type is correct.
AOI specification	rectangular or polygonal AOI specifications are permitted in row,col or lat,lon system (see Appendix 4); for polygonal AOI the surrounding rectangular AOI is used	optional parameter; if not present, the entire input image is assumed
Output Image	the name of the output image containing floating point data (an extension “IFf” is automatically added by the system) Example: Output Image = "float_img"	mandatory OUTPUT BEST extension: IFf

Ancillary Data Dump

Description

The ancillary data dump creates an ASCII file containing a listing of the image annotations which are stored in the specified internal format image. The listing includes the following fields:

a line progressive index
the annotation name
the annotation value

and is a fast way to check the value of the image annotations.

The SAR Toolbox system maintains the entire annotation set only for the header file coming from the extraction tools. The remaining tools maintain only the annotations needed (which are considerably less, due to the fact that the most part of the original CEOS or MPHSPH are not used for the SAR Toolbox processing) with the exception of the TIFF and the BIL conversion function which cut all the annotations from the output file.

Example "INI" file

[ANCILLARY DATA DUMP]
Input Dir = "./"
Output Dir = "./"
Input Image = "cfvr.LSf"
Output File = "dump"

Note: The extension .LSf for the Input Image is just one possible example of a file in the internal SAR Toolbox format.

Ancillary Data Dump Summary Table

Parameter	Description Example	Comment
Input Image	the input image Example: Input Image = "cfvr.LSf"	mandatory INPUT BEST extension: ??i, ??f, ??c, ??s, ??t, ??r where the "??" indicates it is not important which module created the files, as long as the data type is correct.
Output File	the name of the ASCII file containing the annotation listing (an extension “txt” is added by the system) Example: Output File = "dump"	mandatory OUTPUT BEST extension: txt

A sample of a file obtained as output from this function is shown in Appendix 3, together with a table to explain the set of the annotations that are maintained by the various STB tools.

Image Operation

Description

The Image Operation function allows a set of basic algebraic operations to be performed between two images or between one image and a constant factor. The basic operations are sum, subtract, multiply and divide. It is also possible to calculate the absolute value of one image.

Some example INI files are shown here. The following is used for summing two input images:

```
[IMAGE OPERATION]
Input Images = "imagein1", "imagein2"
Operation Type = "SUM"
Output Image = "imageout"
```

This INI file makes it possible to multiply one image with a constant:

```
[IMAGE OPERATION]
Input Images = "imagein"
Operation Type = "MUL"
Output Image = "imageout"
Constant Factor = 1.7
```

Finally, this INI file makes it possible to obtain the absolute value of one image:

```
[IMAGE OPERATION]
Input Images = "imagein"
Operation Type = "ABS"
Output Image = "imageout"
```

The parameter ‘Input Images’ is mandatory and is used to specify one (at least) or two (and no more) input images.
The allowed values for the mandatory parameter ‘Operation Type’ are:
“SUM”, to sum two images, or to sum one image with a constant,
“SUB“, to subtract the second defined image from the first defined image, or to subtract a constant from one image,
“MUL”, to multiply two images or to multiply one image with a constant,
“DIV”, to divide the first defined image by the second defined image, or one image by a constant.
“ABS” to have the absolute value of one image.
The ‘Output Image’ parameter is mandatory and is used to specify the output image.
The ‘Constant Factor’ allows to specify the value for summing, subtracting, multiplying or dividing one image with a constant.

The allowed types for input images are:

8-bits Unsigned Integer (??i), for 8-bit TIFF images,
Complex 8-bits Unsigned Integer (??r), for RAW products,
16-bits Unsigned Integer (??s), for PRI/GEC/GTC products,

Complex 16-bits Signed Integer (??t), for SLC products,
32-bits Float (??f), for internal format files
Complex 32-bits Float (??c), for internal format files

The output image has the extension ‘OP?’: the type ? (float or complex) depends on the input image type.

The following table for shows the type of the output image, when one of the “SUM”, “SUB“, “MUL” and “DIV” operators is applied to two input images:

	??i	??r	??s	??t	??f	??c
??i	OPf	OPc	OPf	OPc	OPf	OPc
??r		OPc	OPc	OPc	OPc	OPc
??s			OPf	OPc	OPf	OPc
??t				OPc	OPc	OPc
??f					OPf	OPc
??c						OPc

Note that in case of operations between a real image (??i, ??s and ??f) and a complex one (??r, ??t and ??c), the real image is considered as a complex image having the imaginary part set to 0.

The type of the output image resulting from one of “SUM”, “SUB“, “MUL” and “DIV” operators applied to one input image and a constant (which is of float type) depends on the type of the input image, as shown in the following table.

	Constant
??i	OPf
??r	OPc
??s	OPf
??t	OPc
??f	OPf
??c	OPc

In case of operations between one complex image (??r, ??t and ??c) with a constant, the constant value is considered as a complex number having the imaginary part set to 0.

The output image type of “ABS” operation is always of float type (OPf).

Selecting an Area of Interest:

An area of interest can be selected with the usual SAR Toolbox rules (Coordinate System, Top Left Corner, Bottom Right Corner, etc. parameters), except for polygonal AOI (Number of Vertex, Vertex couples) which are not allowed (see Appendix 4 for further details about AOI specification).
When executing the absolute value of one image or when adding, subtracting, multiplying or dividing one image by a constant, the size of the resulting output image is given by the selected AOI of the input image, or corresponds to the whole input image size, if no AOI is selected.

The behaviour is different when adding, subtracting, multiplying or dividing two input images. If no AOI is given, the input images must have the same size. This will also be the size of the output image.

When an AOI is given, using this parameter settings

Coordinate System = "LATLON"
Top Left Corner = 41.160, 14.830
Bottom Right Corner = 41.130, 14.800

or the following ones

Coordinate System = "ROWCOL"
Top Right Corner = 100, 599
Bottom Left Corner = 499, 100

The AOI is always interpreted as the starting row, starting column, ending row and ending column of the first input image (the master image). This same AOI (using the starting row, starting column, ending row and ending column resulting from the master image) is applied to the other image (the slave image). A check is made to ensure that the dimensions of the second (slave) image are sufficient to contain the selected AOI.

In general, it is suggested to always use rows and columns to set AOI, because it is more difficult using a latitude and longitude system to ensure that the required area of interest lies within the boundary of the second slave image.

Image Operation Summary Table

Parameter	Description Example	Comment
Input Images	the input image list (one or two images) Example: Input Images = "real1.XTs","real2.XTs"	mandatory INPUT BEST extension: ??i, ??r, ??s, ??t, ??f, ??c
AOI specification	see Appendix 4	optional parameter; if not present, the entire input

		image is assumed
Operation Type	the type of operation selected between: SUM SUB MUL DIV ABS Operation Type = “MUL”	mandatory parameter
Constant Factor	a floating number to be used as constant in the selected operation (except for “ABS”)	optional parameter, needed if only one input image is given and the operation type is not “ABS”
Output Image	the name of the output file containing the resulting image (an extension “OPf” or “OPc” is automatically added by the system) Example: Output Image = "float_img"	mandatory OUTPUT BEST extension: OPf or OPc

Geometric Conversion

Description

The GEOMETRIC CONVERSION tool computes equivalent image coordinates in a selection of imaging geometry reference conventions for specified points in an image. The tool is capable of converting:

- from (row, column) pairs to (latitude, longitude) pairs
- from (latitude, longitude) pairs to (row, column) pairs
- from (row, column) pairs to (incidence angle, look angle) pairs (only for ERS products except GEC and GTC)
- from (row) to (satellite position) expressed in the Earth-centred XYZ system of the orbital state vectors (only for ERS products except GEC and GTC)

All the transformations are computed using the ancillary data contained in the reference image.

The output is a text file indicating the reference image whose ancillary data was used and a list of the original coordinates requested for conversion with their computed equivalents. An “Out Of Image” flag is added to each case where the requested input coordinates (in row, column or latitude, longitude) are outside the limits of the reference image.

Clearly, no AOI can be specified in this operation.

Reasonable checks are executed for verifying the compatibility of ‘Input Coordinates Type’ and ‘Output Coordinates Type’ parameter values. Also, a warning is generated when the ‘Input Coordinates Type’ parameter value is “ROWCOL” or “LATLON” and the number of ‘Input Coordinate’ parameter values is odd (i.e. not a complete list of coordinate pairs). If just one value is supplied, the Geometric Conversion task stops and reports an error. Otherwise, the last value is ignored.

Example "INI" files

Below are some sample “*.ini*” files and their results for selected conversions:

To convert image row to satellite position:
In this case the tool is used to compute the position of the satellite (expressed in the Earth-centred X, Y, Z system of the orbital state vectors) at a series of points during the acquisition specified by image row numbers. Note that, for demonstration purposes, two of the specified rows lie outside the reference image under consideration (i.e. rows -100 and 1999).

```
[GEOMETRIC CONVERSION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Reference Image = "i09.XTs"
Input Coordinates Type = "ROW"
Input Coordinates = 0, -100, 999, 1999, 100, 250
Output Coordinates Type = "SATPOS"
Output File = "geoconv"
```

Output file “*geoconv.txt*”:

```
================================================================================
       STB - BEST ToolBox - Telespazio / ESA - GEOMETRIC CONVERSION
================================================================================
Reference Image: dat$:i09.XTs
================================================================================
(ROW) -> (X SATELLITE POSITION, Y SATELLITE POSITION, Z SATELLITE POSITION)
(0) -> (5173858.256349, 1665150.608019, 4639795.873503)
(-100) -> (5172906.319468, 1665206.232858, 4640834.441325) *** OUT OF IMAGE ***
(999) -> (5183357.284946, 1664590.037330, 4629410.891974)
(1999) -> (5192846.092677, 1664020.027111, 4618997.901304) *** OUT OF IMAGE ***
(100) -> (5174809.996228, 1665094.894292, 4638757.129063)
(250) -> (5176237.236584, 1665011.157048, 4637198.681331)

================================================================================
```

To convert (row, column) to (latitude, longitude):
In this case the tool is used to compute latitude, longitude pairs from a list of row, column pairs. Note that, for demonstration purposes, two of the specified points lie outside the reference image under consideration.

```
[GEOMETRIC CONVERSION]
...
Reference Image = "i09.XTs"
Input Coordinates Type = "ROWCOL"
Input Coordinates = 0, 0, -100, 0, 999, 999, 1999, 0, 100, 250, 150, 250
Output Coordinates Type = "LATLON"
Output File = "geoconv"
```

Output file "*geoconv.txt*":

```
================================================================================
       STB - BEST ToolBox - Telespazio / ESA - GEOMETRIC CONVERSION
================================================================================
Reference Image: dat$:i09.XTs
================================================================================
(ROW, COLUMN) -> (LATITUDE, LONGITUDE)
(0, 0) -> (41.188416, 14.906085)
(-100, 0) -> (41.199415, 14.909317) *** OUT OF IMAGE ***
(999, 999) -> (41.102253, 14.728530)
(1999, 0) -> (40.968536, 14.841642) *** OUT OF IMAGE ***
(100, 250) -> (41.183374, 14.866438)
(150, 250) -> (41.177875, 14.864826)

================================================================================
```

To convert (row, column) to (incidence, look angle):
In this case the tool is used to compute incidence angle and look angle for a series of row, column pairs. Note that, for demonstration purposes, two of the specified points lie outside the reference image under consideration.

```
[GEOMETRIC CONVERSION]
...
Reference Image = "i09.XTs"
Input Coordinates Type = "ROWCOL"
Input Coordinates = 0, 0, -100, 0, 999, 999, 1999, 0, 100, 250, 150, 250
Output Coordinates Type = "INCLOK"
Output File = "geoconv"
```

Output file "*geoconv.txt*":

```
================================================================================
       STB - BEST ToolBox - Telespazio / ESA - GEOMETRIC CONVERSION
```

```
================================================================================
Reference Image: dat$:i09.XTs
================================================================================
(ROW, COLUMN) -> (INCIDENCE ANGLE, LOOK ANGLE)
(0, 0) -> (20.273800, 17.952206)
(-100, 0) -> (20.273291, 17.951748) *** OUT OF IMAGE ***
(999, 999) -> (21.202480, 18.768332)
(1999, 0) -> (20.283932, 17.961334) *** OUT OF IMAGE ***
(100, 250) -> (20.506369, 18.156677)
(150, 250) -> (20.506618, 18.156904)

================================================================================
```

To convert (latitude, longitude) to (row, column):
Finally, the tool is used to compute the row, column location of selected latitude, longitude geographic coordinate pairs. Note that, for demonstration purposes, one of the specified locations lies outside the reference image under consideration.

```
[GEOMETRIC CONVERSION]
...
Reference Image = "i09.XTs"
Input Coordinates Type = "LATLON"
Input Coordinates = 41.188416, 14.906085, 41.102253, 14.728530, 41.183374,
14.8, 41.0, 16.0
Output Coordinates Type = "ROWCOL"
Output File = "geoconv"
```

Output file "*geoconv.txt*":

```
================================================================================
       STB - BEST ToolBox - Telespazio / ESA - GEOMETRIC CONVERSION
================================================================================
Reference Image: dat$:i09.XTs
================================================================================
(LATITUDE, LONGITUDE) -> (ROW, COLUMN)
(41.188416, 14.906085) -> (0, 0)
(41.102253, 14.728530) -> (999, 999)
(41.183374, 14.866438) -> (100, 250)
(41.177875, 14.864826) -> (150, 250)
(41.000000, 16.000000) -> (31, -7540) *** OUT OF IMAGE ***

================================================================================
```

Parameter Summary: Geometric Conversion

Reference Image

The name of the integer image in internal format for which conversions will be computed.
Example: `Reference Image = "t1_priimage.XTs"`
mandatory INPUT
BEST extension: ".??i", ".??f", ".??c", ".??s", ".??t", ".??r" where "??" indicates that any BEST module could have produced this file.

Input Coordinates Type

The type of input coordinates to be transformed:

- "ROWCOL" (row, column pairs)
- "LATLON" (latitude, longitude coordinates)
- "ROW" (list of image rows)

Example: `Input Coordinates Type = "ROW COL"`
mandatory parameter

Input Coordinates

A list of comma-separated (row, column) or (latitude, longitude) pairs or a list of comma-separated image row numbers (depending on the 'Input Coordinates Type' parameter) to be converted.

Example: `Input Coordinates = 0, 0, 999, 999, 100, 250, 150, 250`

mandatory parameter

Output Coordinates Type

The type of output to be computed:

- "LATLON" (row, column pairs)
- "ROWCOL" (latitude, longitude coordinates)
- "INCLOK" (incidence angle and look angle)
- "SATPOS" (satellite position X, Y, Z triplets)

Example: `Output Coordinates Type = "LATLON"`

mandatory parameter

Output File

The name of the output file containing the computed conversions (an extension "*.txt*" is automatically added by the system).

Example: `Output Image = "float_img"`

mandatory OUTPUT

BEST extension: ".txt"

Slant Range to Ground Range Conversion

Description

The SLANT TO GROUND RANGE CONVERSION tool reprojects data (complex as SLC products in internal format or co-registered images, or real as coherence images) from slant range onto a flat ellipsoid surface.

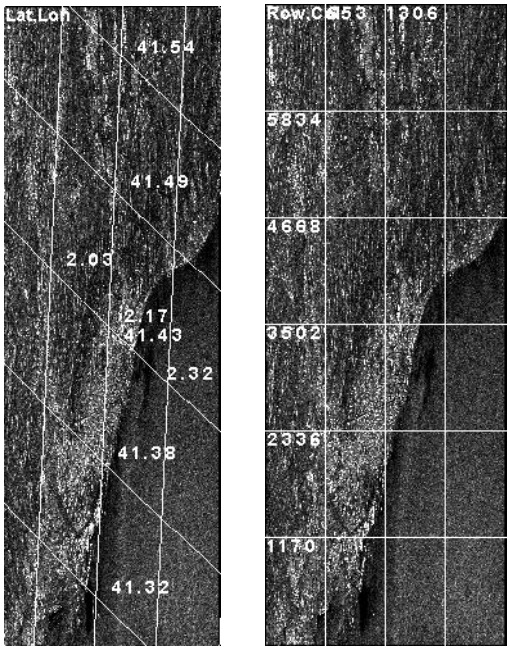
The following steps are implemented:

- construction of a regular interpolation axis in ground range by coordinate conversion from (row,col) to (x,y,z)
- evaluation of a set of ground range to slant range polynomial coefficients (having a fixed degree) for a fixed number of rows of the slant range image
- evaluation of the corresponding ground range values from slant range-azimuth using the previously evaluated coefficients
- range interpolation of the image data with the cubic convolution interpolator.

The cubic convolution interpolator uses five interpolations with a four-coefficient cubic convolution kernel applied to the sixteen pixels around the position determined by the transformation function.

The tool makes no adjustment to the data in the azimuth direction. Therefore, an elongated single-look complex image will remain a single-look image. The purpose of the SLANT TO GROUND RANGE CONVERSION tool is to redistribute the data in range with equal pixel spacing. See below for the full sequence of processing necessary in BEST to convert SLC data to a multi-looked output image.

The figures below show quick look images with superimposed latitude–longitude grids of an IMS image of Barcelona before and after slant to ground range conversion.



(left) Quick look of an SLC image over Barcelona (2377 columns);
(right) Quick look of the ground-projected image (2614 columns, same number of rows)

Typical Processing Chain

The SLANT TO GROUND RANGE CONVERSION tool may be used as part of a processing chain to generate a multi-looked (PRI-like) image in ground range starting from an SLC product.

Several steps are necessary to perform multi-looking on an SLC image, as described below:

- oversampling (2 × 2) in SLC format - zero-padding avoids aliasing problems
- slant-to-ground range re-projection in SLC format
- look detection (amplitude image)
- look adding by undersampling with the desired multi-look factor

The corresponding processing chain in terms of tools in BEST would be:

HEADER ANALYSIS ⇒ FULL RESOLUTION ⇒ OVERSAMPLING ⇒ SLANT TO GROUND RANGE CONVERSION ⇒ COMPLEX TO AMPLITUDE CONVERSION ⇒ UNDERSAMPLING ⇒ EXPORT

The OVERSAMPLING ‘Output Image Ratio’ should be “2, 2”.
The UNDERSAMPLING ‘Output Image Ratio’ should be “0.166, 0.5” for a 3-look output.

However, to improve computational performance, it is advised to apply slant-to-ground range re-projection to the amplitude image by rearranging the processing chain thus:

HEADER ANALYSIS ⇒ FULL RESOLUTION ⇒ OVERSAMPLING ⇒ COMPLEX TO AMPLITUDE CONVERSION ⇒ UNDERSAMPLING ⇒ SLANT TO GROUND RANGE CONVERSION ⇒ EXPORT

Important: Oversampling a full SLC scene from ERS or Envisat will increase the file size above the maximum (2Gb) allowed for TIFF handling. It is recommended that this processing chain is performed only for sub-scenes.

Example "INI" file

```
[SLANT RANGE TO GROUND RANGE CONVERSION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "asar_ims.XTt"
Output Image = "sl2gr"
Delete Input Image = "N"
```

Parameter Summary: Slant Range to Ground Range Conversion

Input Image

The complex or real image in slant range to be reprojected
Example: `Input Image = "asar_ims.XTt"`
mandatory INPUT
BEST extension: ".XTt", ".CRc", ".CHf", ".ML", ".CAf", ".APf", ".OPc", ".OPf", ".BSf" or ".GAf"

AOI specification

see Appendix 4; for polygonal AOIs the surrounding rectangular AOI is used
optional parameter (default is entire input image)

Output Image

The name of the output ground-projected image (an extension "*.SGf*" or "*.SGc*" is automatically added by the system)
Example: `Output Image = " sl2gr "`
mandatory OUTPUT
BEST extension: ".SGf" or ".SGc"

Flip Image

Description

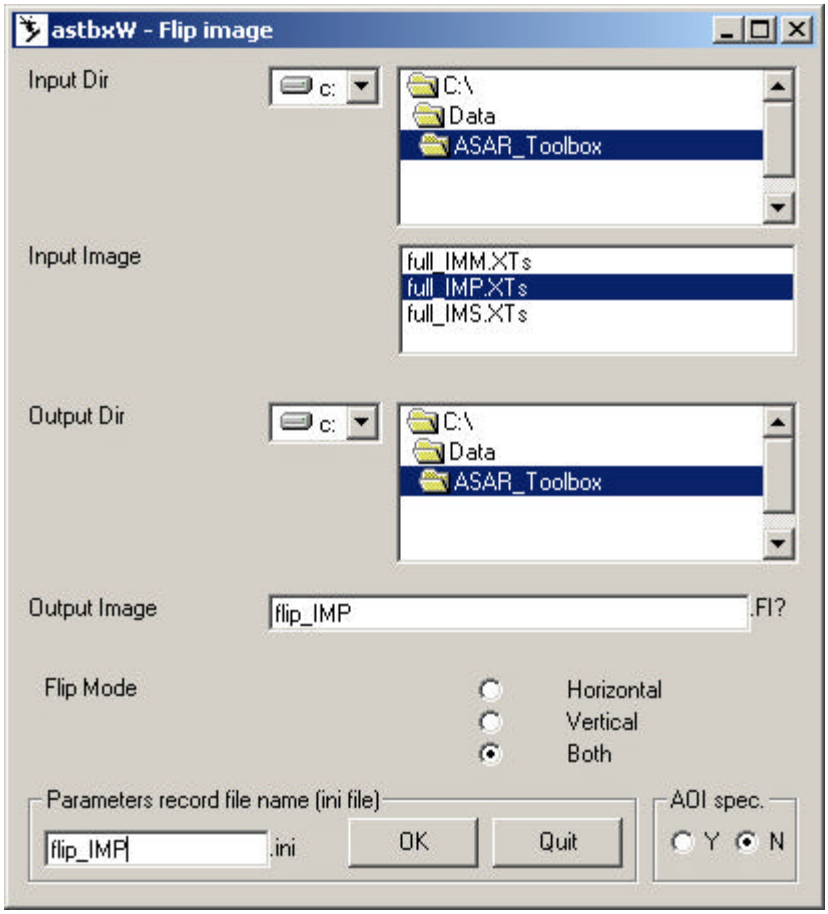
The FLIP IMAGE function performs a simple affine transformation on an image in the internal Toolbox format, to render it in a recognisable form without running the Geo-correction tool.

The ASAR Toolbox automatically locates the first pixel of the first line of a data set in the top left corner of the image. In reality, the first data sample is located in the bottom left corner of the scene for ascending passes and the top right corner for descending passes.

By applying a vertical or horizontal flip, the image can be oriented so that north is up, south is down, west is left and east is right.

The flip is based on the row, col reference system and can be executed with respect to the vertical axis, the horizontal axis or to both at the same time.

HMI



Typical HMI settings for an ASA_IMP_1P product

Typical Processing Chain

HEADER ANALYSIS ⇒ FULL RESOLUTION ⇒ FLIP IMAGE

Example "INI" file

```
[FLIP IMAGE]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "asar_img.XTs"
Delete Input Image = "N"
Output Image = "vflp_img"
Flip Mode = "vertical"
```

Parameter Summary: Flip Image

Input Image

The internal format image to flipped.
Example: `Input Image = "asar_img.XTs"`
mandatory INPUT
BEST extension: ".??i", ".??s", ".??t" , ".??f" or ".??c" where "??" indicates that any BEST module could have produced these files.

AOI specification

see Appendix 4; for polygonal AOI the surrounding rectangular AOI is used
optional parameter (default is entire input image)

Output Image

The name of the output flipped image (the extension "*.FI?*" is automatically added by the system)
Example: `Output Image = "vflp_img"`
mandatory OUTPUT
BEST extension: ".FIf", ".FIc"

Flip Mode

The sense of the flip operation:

- "VERTICAL"
- "HORIZONTAL"
- "BOTH"

Example: `Flip Mode = "vertical"`
mandatory parameter

Sensitivity Vector Evaluation

Description

The SENSITIVITY VECTOR EVALUATION tool calculates the sensitivity vector of an input image point by point. The user may specify points one by one or alternatively define an equally spaced grid by its row and column dimensions.

The displacement measurable by radar using interferometry is the projection of a real displacement into the radar line of sight (z). If the displacement at a point, d(x,y) is represented in a local orthogonal frame of reference (North, East, Vertical) by 3 components:

$$d = (d_N, d_E, d_V)$$

and the LOS between the radar and that point is represented in the same reference frame by the vector:

$$z = (z_N, z_E, z_V)$$

the measured displacement at the point (x,y) is given by the following scalar product:

$$D(x,y) = d(x,y) ? z(x,y)$$

The vector, z, is called the sensitivity vector; it gives a measure of the sensitivity of the measurement of displacement in each of three orthogonal axes. For a given radar acquisition, it varies with latitude and longitude within the scene. A typical value for ERS would be z = (0.01, 0.3, 0.9). Hence:

$$D(x,y) = 0.01\ d_N, + 0.3\ d_E, + 0.9\ d_V$$

In this case, D(x,y) is principally a measure of the vertical component of a real displacement, with a contribution from any movement in the East-West. Sensitivity to movement in the North-South direction is negligible. The sensitivity vector is determined by the incidence angle and orbital inclination of a radar system. By increasing the incidence angle (possible using image swaths 3 to 7 of Envisat ASAR), the sensitivity to the East-West component of a ground displacement increases whilst the sensitivity to the vertical component decreases.

The following example shows the format of an output "*.txt*" file for a point defined by lat,lon. For each input (row,col or lat,lon) point, the local east, north and vertical components of the sensitivity vector are reported in metres.

```
================================================================================
BEST - ESA / Telespazio - SENSITIVITY File
================================================================================
 | Lat          | Lon          | Sn [m]       | Se [m]       | Sv [m]       |
================================================================================
-|--------------|--------------|--------------|--------------|--------------|
0|   36.50000000| -114.59999847|  -0.347267029|  -0.347259363|  -0.001754421|
```

Example "INI" file

```
[SENSITIVITY EVALUATION]
```

```
Input Dir = "G:\"
Output Dir = "G:\"
Input Image = "img.XTs"
Output File = "sns_vect"
Input Coordinates Type = "POINTGRID"
Number of Points = 10
```

Parameter Summary: Sensitivity Evaluation

Input Image

The name of the internal format input image.
Example: `Input Image = "img.XTs"`
Mandatory INPUT
BEST extension: any internal Toolbox format file

Output File

The name of the output file containing the values of the sensitivity vector calculated for the specified points. The extension "*.txt*" is automatically added by the system.
Example: `Output File = "sns_vect"`
Mandatory OUTPUT
BEST extension: ".txt"

Input Coordinates Type

The manner in which points shall be defined:

- "ROWCOL" (one by one, by rows and columns)
- "LATLON" (one by one, by latitude and longitude)
- "POINTGRID" (define an equally spaced grid)

Example: `Input Coordinates Type = "POINTGRID"`
Mandatory INPUT

Number of Points

The dimensions, expressed in number of rows and columns, of the equally spaced grid of points at which the sensitivity vector will be calculated, to be automatically generated where 'Input Coordinates Type' is "POINTGRID".
Example: `Number of Points = 10`
optional parameter

Input Coordinates

The coordinates, in row,col or lat,lon, of the point(s) at which the sensitivity vector will be calculated where 'Input Coordinates Type'is "ROWCOL" or "LATLON".
Example: `Input Coordinates = 45.0, 15.0`
optional parameter

10. Statistical

This chapter documents the following tools:

1. Global Statistic
Calculates a range of statistical parameters (mean, standard deviation, coefficient of variation, equivalent number of looks) for an image or region of interest within an image. Also generates a histogram of the pixel values.

2. Local Statistic
Generates output images showing a range of statistical parameters (mean, standard deviation, coefficient of variation, equivalent number of looks) computed from an image using a moving window of selectable size.

3. Principal Components Analysis
Generates the first and second principal components from a pair of input images.

Global Statistic

Description

The GLOBAL STATISTIC function computes some statistical parameters for an image or area of interest (AOI) within an image. The statistical parameters are the standard deviation, coefficient of variation, equivalent number of looks, mean value, image maximum, image minimum and a histogram of the pixel values. These values are global, i.e. one unique value of a certain statistical parameter is given for the entire AOI.

The AOI can be specified in any way, except the example image mode.

Example "INI" file

```
[GLOBAL STATISTIC]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "t1_priimage.XTs"
Top Left Corner = 100, 100
Bottom Right Corner = 700, 700
Class Min = 100.0
Class Max = 900.0
Classes Number = 8
Output File = "glostat"
```

Parameter Summary: Global Statistic

Input Image

The name of the input real image in internal format.
Example: `Input Image = "t1_priimage.XTs"`
mandatory INPUT
BEST extension: “.??s”, “.??i”, “.??f” where "??" indicates it is not important which module created the files, as long as the data type is correct.

AOI specification

see Appendix 4
optional parameter (default is entire input image)

Class Min

A floating point number specifying the second class of the histogram; image pixels having a value lower or equal to this shall all contribute to the first histogram bin.
Example: `Class Min = 100.0`
mandatory parameter

Class Max

A floating point number specifying the penultimate class of the histogram; image pixels having a value greater or equal to this shall all contribute to the last histogram bin.
Example: `Class Max = 900.0`
mandatory parameter

Classes Number

An integer specifying the number of classes in the histogram (minus 2); the histo gram shall contain two class more than this number, the first for all the pixels having a value below ‘Class Min’ and the last for all the pixels having a value greater than ‘Class Max’.
Example: `Classes Number = 8`
mandatory parameter

Output File

The name of the output ASCII file containing the global statistical data (the extension “*.txt*” is automatically added by the system)
Example: `Output File = "glostat"`
mandatory OUTPUT
BEST extension: “.txt”

Local Statistic

Description

The LOCAL STATISTIC function computes a statistical parameter within a (usually small) window (kernel) that is allowed to move across an image. The statistical parameter that is computed for each position of this kernel is used to build-up an output image that presents information about the local statistics of the input image.

The statistical parameters available are the mean, standard deviation, coefficient of variation and equivalent number of looks.

It is possible for the user to specify the size of the kernel in which the statistical parameters are calculated. It is also possible for the user to specify the step size that determines the frequency with which the statistical parameter is calculated. These step sizes will also determine the size of the output image.

The user can also define the output image size by specifying the ‘Output Image Ratio’ values along the rows and columns (see the second example "INI" file below).

The area of interest (AOI) of the input image can be specified in any way (except the example image mode), including polygonal AOI. When the kernel is partly or completely outside the AOI no statistics are generated. For more details see the ‘Statistical tools’ chapter of the Algorithm Specification Document [A3].

Example "INI" files

```
[LOCAL STATISTIC]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.XTs"
Filler Value = 0.0
Coordinate System = "ROWCOL"
Top Left Corner = 100, 100
Bottom Right Corner = 500, 500
Output Image Type = "MEAN"
Window Sizes = 5, 5
Window Steps = 2.0, 2.0
Output Image = "t1_locstatmean"
```

In the following example the output image size is determined by the parameter ‘Output Image Ratio’. The output image “local.LSf” will have (299-100+1)*0.5 rows and (399- 100+1)*0.7 columns:

```
[LOCAL STATISTIC]
Input Dir = "./"
Output Dir = "./"
Input Image = "pri.XTs"
Output Image = "local"
Output Image Type = "MEAN"
Output Image Ratio = .5, .7
Window Sizes = 3, 5
Top Left Corner = 100, 100
Bottom Right Corner = 299, 399
```

Local Statistic Summary Table

Input Image

The name of the input real image in internal format.
Example: `Input Image = "t1_priimage.XTs"`
mandatory INPUT
BEST extension: “.??s”, “.??i”, “.??f” where "??" indicates it is not important which module created the files, as long as the data type is correct.

AOI specification

see Appendix 4
optional parameter (default is entire input image)

Output Image Type

The type of local statistical operation:

- “MEAN” (moving mean)
- “SDDV” (moving standard deviation)
- “CFVR” (moving coeffi cient of variation_
- “ENLV” (moving equivalent number of looks)

Example: `Output Image Type="MEAN"`
mandatory parameter

Window Sizes

The size of the kernel used to compute the local statistic; a couple of integer numbers comma separated, the first one referring to the number of rows, the second to the number of columns.
Example: `Window Sizes = 5, 5`
mandatory parameter

Output Image Ratio

The ratio by which the dimensions of the input image are transformed in the output image. This parameter is an alter native to the ‘Window Steps’ parameter.
Example: `Output Image Ratio = .5, .7`

Window Steps

The rate at which the local statistic kernel is moved, set to a value different from 1 to sub sample the corresponding statistic image; e.g. when computing a local statistic on a 100 by 100 image with a 10 by 10 kernel and a 20 by 20 step, the output image will be only 5 by 5 pixels; a step of 1 by 1 would generate a full image 91 by 91 pixels (less than 100 by 100 because of the “kernel edge” effect). This parameter is an alternative to the ‘Output Image Ratio’ parameter.
Example: `Window Steps = 2.0, 3.0`

Filler Value

Should a polygonal AOI be used, this specifies the value to be assigned to pixels in the output image (always rectangular) which do not fall within it.
Example: `Filler Value = -1.0`
optional parameter (default is “0”)

Output Image

The name of the output image in internal format containing the local statistic image.
Example: `Output File = "local_mean"`
mandatory OUTPUT

BEST extension: “.LSf”

Principal Components Analysis

Description

The PRINCIPAL COMPONENT ANALYSIS tool generates the first and second principal component images from two input images. The output images are scaled to avoid negative pixel values.

The AOI may only be defined by the rectangular method, with corners expressed in row,col.

Example "INI" file

```
[PRINCIPAL COMPONENT ANALYSIS]
Input Dir = "./"
Output Dir = "./"
Input Images = "img1.XTf", "img2.XTf"
PCA Output Images = "pc1", "pc2"
```

Parameter Summary: Principal Component Analysis

Input Images

The names of two input real images in internal format, in a comma separated list.
Example: `Input Images="img1.XTf", "img2.XTf"`
mandatory INPUT
BEST extension: “.??s”, “.??i”, “.??f”, where "??" indicates it is not important which module created the files, as long as the data type is correct.

AOI specification

see Appendix 4; only the rectangular method, with corners expressed in row,col may be used.
optional parameter (default is entire input image)

PCA Output Images

The names of the output images in internal format containing the first and second Principal Components (the extension “*.PCf*” is automatically added by the system), in a comma separated list.
Example: `PCA Output Images="pc1","pc2"`
mandatory OUTPUT
BEST extension: “.PCf”

11. Resampling

This chapter documents the following tools:

1. Oversampling (Up-Sampling)
Resamples an image to increase the number of pixels.

2. Undersampling (Down-Sampling)
Resamples an image to reduce the number of pixels.

Oversampling (Up-Sampling)

Description

The Oversampling function “up-samples” a real or complex image (like a SLC product) using the FFT and Zero Pad algorithm. The algorithm takes into account the value of the Doppler Centroid Frequency when padding the azimuth spectrum.

The size of the output image can either be determined by specifying this directly in terms of the pixel dimensions, e.g. Output Image Size. Or, in terms of the row, col ratios by which the input image should be multiplied. These numbers should be greater than 1 (see the second example "INI" file below).

Only the rectangular AOI with corners specified in row, col system, are accepted.

Example "INI" files

The following “*.ini*” file is an example for the oversampling for an input SLC image portion.

[IMAGE OVERSAMPLING]
Input Dir = "./"
Output Dir = "./"
Input Image = "slcimage.XTt"
Output Image = "oversam"
Top Left Corner = 100,100
Bottom Right Corner = 199,299
Output Image Size = 200,300

The output from the following "INI" file would have dimensions (299-100+1)*1.5 rows and (399- 100+1)*1.1 columns.

[IMAGE OVERSAMPLING]
Input Image = "priimage.XTs"
Output Image = "oversam"
Top Left Corner = 100,100
Bottom Right Corner = 299,399
Output Image Ratio = 1.5,1.1

Oversampling (Up-Sampling) Summary Table

Parameter	Description Example	Comment
Input Image	the name of the input real or complex image in internal format Example: Input Image = "slcimage.XTs"	mandatory INPUT BEST extension: ??i, ??f, ??c, ??s, ??t, ??r where the "??" indicates it is not important which module created the files, as long as the data type is correct.

AOI specification	only the Rectangular AOI specifications are permitted (see Appendix 4)	optional parameter; if not present, the entire input images are assumed
Output Image Size	the size of the output image, expressed as number of rows, number of columns; these sizes shall be greater than the input image size; this number permits to control the over sampling factor Output Image Size=2000,1500	This parameter is an alter native to Output Image Ratio
Output Image Ratio	The row, col ratios to determine the size of the output file. These numbers should be greater than 1. Output Image Ratio = 1.5,1.1	This parameter is an alter native to Output Image Size
Output Image	the name of the output image in internal format containing the oversampled image (an extension “OVf or OVc” is automatically added by the system) Example: Output Image = "oversam"	mandatory OUTPUT BEST extension: OVf or OVc

Undersampling (Down-Sampling)

Description

The undersampling function down-samples a real image (like a PRI or GEC product) using a kernel, which moves across the input image with a step-size determined by the size of the required output image.

The size of the output image can either be determined by specifying this directly in terms of the pixel dimensions, e.g. Output Image Size. Or, in terms of the row, col ratios by which the input image should be multiplied. These numbers should be less than 1 (see the second example "INI" file below).

The kernel can be selected for one of a set of predefined kernels or can be defined by the user. A list of the pre-defined kernels is given here after the ‘example "INI" file’.

The kernel file is a simple ASCII file containing rows of coefficients separated by spaces, with each row terminated with a newline (return) character.
There are two modes in which the kernel files can be used:

2D: One method allows 2-dimensional filters to be defined directly. In this case an example of an ASCII file for a 3x3 averaging filter would be:

0.1111 0.1111 0.1111
0.1111 0.1111 0.1111
0.1111 0.1111 0.1111
This particular example is contained in the pre-defined file usam33.ker.

2 x 1D: The other method makes use of two 1-dimensional filters to effectively synthesise a 2-dimensional filter. This method involving two 1-dimensional filters can be much faster than the direct use of a 2-dimensional filter. For example, for a 11 rows by 11 columns filter, this filtering can be five times faster than a conventional 2-dimensional filtering.

The two 1-dimensional filters that would produce the equivalent filter to the 3x3 averaging filter shown above would be defined by the following ASCII file:

0.3333 0.3333 0.3333
0.3333
0.3333
0.3333

The SAR Toolbox system automatically discovers the correct way to interpret the kernel depending on the layout of the ASCII file.

As another example, consider the following 4x3 filter:

1.0 2.0 3.0 4.0
2.0 4.0 6.0 8.0
3.0 6.0 9.0 12.0

This could be synthesised by two 1-dimensional filters described by the following ASCII representation:
1.0, 2.0, 3.0, 4.0
1.0
2.0
3.0

Example "INI" file

The following "INI" file shows the function working on a PRI image with an output image size defined in terms of pixels.

```
[IMAGE UNDERSAMPLING]
Input Dir = "./"
Output Dir = "./"
Filter File Name = "usam33.ker"
Input Image = "priimage.XTs"
Top Left Corner = 100, 200
Bottom Right Corner = 399, 599
Output Image = "undersam"
Output Image Size = 200, 200
```

The following "INI" file shows the function working on a PRI image with an output image size defined in terms of ratios.

```
[IMAGE UNDERSAMPLING]
Input Dir = "./"
Output Dir = "./"
Filter File Name = "usam33.ker"
Input Image = "pri.XTs"
Top Left Corner = 100, 100
Bottom Right Corner = 299, 399
Output Image = "undersam"
Output Image Ratio = .5, .7
```

This would give an output file of size (299-100+1)*0.5 rows and (399-100+1)*0.7 columns.

The following table describes the library of preset kernels.

Kernel file name	Comment
edd_3_3.ker	3x3 Edge Detect
ede_3_3.ker	3x3 Edge Enhance
lop_3_3.ker	3x3 Low Pass
hip_3_3.ker	3x3 High Pass
hor_3_3.ker	3x3 Horizontal
ver_3_3.ker	3x3 Vertical
sum_3_3.ker	3x3 Summary
edd_5_5.ker	5x5 Edge Detect
ede_5_5.ker	5x5 Edge Enhance
lop_5_5.ker	5x5 Low Pass
hip_5_5.ker	5x5 High Pass
hor_5_5.ker	5x5 Horizontal

ver_5_5.ker	5x5 Vertical
sum_5_5.ker	5x5 Summary
edd_7_7.ker	7x7 Edge Detect
ede_7_7.ker	7x7 Edge Enhance
lop_7_7.ker	7x7 Low Pass
hip_7_7.ker	7x7 High Pass
hor_7_7.ker	7x7 Horizontal
ver_7_7.ker	7x7 Vertical
sum_7_7.ker	7x7 Summary

Undersampling (Down-Sampling) Summary Table

Parameter	Description Example	Comment
Input Image	the name of the real input image in internal format Example: Input Image = "t1_priimage.XTs"	mandatory INPUT BEST extension: ??s, ??i, ??f where the "??" indicates it is not important which module created the files, as long as the data type is correct.
AOI specification	only the Rectangular AOI specifications are permitted (see Appendix 4)	optional parameter; if not present, the entire input image is assumed
Filter File Name	the name of the ASCII file which contains the coefficients of the filter used to subsam ple the input image; the file names of the preset filters are shown in the previous table. Filter File Name="usam55.ker"	mandatory parameter
Output Image Size	the size of the output image, shall be lower than the input image size; this number per mits to control the undersampling factor Output Image Size=20,15	This parameter is an alter native to Output Image Ratio
Output Image Ratio	The row, col ratios to determine the size of the output file. These numbers should be between 0.0 and 1.0. Output Image Ratio = 0.7,0.5	This parameter is an alter native to Output Image Size
Output Image	the name of the output image in internal format containing the undersampled image (an extension “UNf” is automatically added by the system) Example: Output Image = "under sam"	mandatory OUTPUT BEST extension: UNf

12. Co-registration and Coherence Generation

This chapter documents the following tools:

1. Co-registration
Registers one or more images to another using up to three separate processes to achieve a precise fit. Images can be real or complex.

2. Coherence Generation
Calculates the phase coherence between two co-registered complex images.

3. Footprint Registration
Indicates on a quick look of a master image the ‘footprints’ of up to 10 co-registered slaves.

4. Image Geo-correction
Reprojects ASAR medium resolution imagery to a UTM or UPS planar grid.

5. Amplitude-Coherence Multi-layer Composite
Generates a multi-layer pseudo-true-colour composite image consisting of the coherence between two co-registered images with either their mean backscatter and the backscatter difference or the detected images of the master and slave.

6. Doris Baseline Evaluation
Calculates the baseline, based on input DORIS orbit files, between the nearest point of two orbits to a specified ground location.

Co-registration

General

The CO-REGISTRATION tool will co-register one (slave) image, or a set of (slave) images with respect to a (master) image. The function is fully automatic, in the sense that it does not require the user to manually select ‘tie points’ from the master and slave images.

The co-registration is performed in 1, 2 or 3 steps.

1) An initial registration step is performed using the satellite orbit parameters.

2) By default, a coarse registration is achieved using a cross correlation operation on a series of ‘cells’ defined across the images. This step may be disabled by changing the flag parameter ‘Image Coarse Reg’.

3) By default (for complex data), a further fine registration is achieved by the maximization of the complex coherence between the images for a series of ‘cells’ defined across the images, thereby allowing a further improvement on the cross-correlation function. This step can be executed only if the coarse registration (step 2) has been performed. It may be disabled by changing the flag parameter ‘Image Fine Reg’.

Input Images

The input images for the co-registration can be complex (i.e. SLC or SLCI - RAW data is not permitted) or real (i.e. PRI, GEC or GTC). All of the input images must be of the same type (i.e. they must all be complex or all real) and have the same projection system (all slant range or all ground range projected or all geocoded). It is therefore not possible to mix product types, although, the GEC format can been registered with the GTC format because they are both in the east, north projection.

Before performing the main co-registration steps, the tool makes a quick check to ensure the images overlap to a significant degree. If the area of overlap is less than the user configurable threshold, ‘Overlapping AoI Threshold’, the program ends with an error.

Ground Control Points

The coarse and fine co-registration steps operate on rectangular regions within the images defined by a series of ground control points (GCPs). The generation of the GCPs is controlled by one of two methods. Normally, the GCPs are defined automatically on a rectangular grid, but their positions may also be specified by the user in an ASCII file defined using the parameter ‘GCPs File Name’ (see below in the paragraph ‘Specified GCPs’). The number of rows and columns in the automatically-derived grid, defined on the master image, is determined by the parameter ‘GCPs Numbers’ (the total number of GCPs being the product of the two dimensions).

Coarse Registration Step (step 2)

A coarse registration step is performed using a cross correlation operation on a series of ‘cells’ centred on the GCPs (see above) defined across the master image. For both real and complex images the coarse registration step can be switched on or off using the flag parameter ‘Image Coarse Reg’.

The size of the coarse registration cells is defined by the parameter 'Coarse Reg Window Sizes'. In general, the larger is the cell window size, the longer will be the program running time. The accuracy and program running time for the coarse registration step is also determined by the parameter 'Coarse Reg Interp Factors'. This parameter sets the row,col interpolation factors for the coarse registration step. The units are $pixel^{-1}$ and this parameter defines the step size used in the cell cross correlation process. Higher values produce good accuracy at the expense of longer running time.

A check on the accuracy achieved by the coarse registration step is performed for each GCP cell and evaluated with respect to the parameter 'Coarse Reg Tolerance'. The coarse registration step is performed twice, using slightly different cell positions. In the first instance, each cell is centred on a GCP. In the second, the cell centre is defined by the position found using the first step. If the final transformation positions from these two computations do not agree within the limit set by 'Coarse Reg Tolerance' for a particular GCP, it will not be used in the remaining co-registration process. Therefore the parameter (measured in pixel units) provides a check on the stability of the cross correlation procedure.

Fine Registration Step (step 3)

For complex images, the fine registration step is based upon a coherence maximisation routine. This operates on a series of cells defined at the GCPs. For both real and complex images the fine registration step can be switched on or off using the flag parameter 'Image Fine Reg'.

It is possible to define a 'Coherence Threshold' parameter when using the fine co-registration step. GCPs are excluded from the co-registration calculation if their coherence levels fall below this threshold. It is preferable to exclude these low coherence points from the co- registration process because otherwise the calculated translations will essentially be random in areas of very low coherence, for example over regions of water. The 'Coherence Threshold' parameter should have a value between 0 and 1. A sensible value is 0.4.

For real images, the fine registration step involves a further maximisation of the cross correlation function.

The fine registration step makes use of a two-dimensional downhill simplex algorithm. This algorithm is used to search for a maximum in the coherence (or cross correlation for real images). The algorithm uses two possible stopping criteria, which are defined by parameters that can be adjusted by the user.

The first stop criterion is set by the parameter 'Coherence Func Tolerance'. When the change in the coherence, produced after a given cycle of the algorithm, is below this tolerance, the search stops. The second stop criterion is set by the parameter 'Coherence Value Tolerance'. When the shifts (in units of pixels) made on the slave cell are below this tolerance, the search stops.

For real data, the size of the fine registration cells is defined by the parameter 'Fine Reg Window Sizes'. In general, the larger the cell window size is, the longer the program running time will be.

For complex data, the size of the square window in which the coherence is calculated (within the GCP cells) is determined by the parameter 'Coherence Window Size'. Small values of this parameter can lead to high noise levels in the coherence calculation. A value of 7 or 9 is recommended.

Excluding the Worst GCPs

It is possible to select only the best GCPs by excluding those that cannot be properly fitted by a polynomial warping function. This is achieved by making an initial generation of the warp function using all of the GCPs and then excluding the worst GCPs (i.e. those which generate the highest residuals). In this way only a set of GCPs which are compatible with the polynomial warp function are selected. This operation is controlled by the parameter 'Editing RMS'. If the residual for a particular GCP is greater than the defined threshold, it will be discarded.

Specified GCPs

In some cases it can be useful for the user to select GCPs on the master image manually. This is possible by providing a text file (defined by the parameter 'GCPs File Name') containing the GCP coordinates from the master image in the row,col coordinate system. (Obviously, there is no need to specify the slave positions of the GCPs as these are computed by the system.) This option is useful for registering images that contain large regions of low coherence (e.g. an image of a coastal area). If the GCPs are not specified directly, then they will be uniformly distributed in the image and it may be the case that only a few of them are placed in areas of sufficient coherence. In contrast, by using the GCP file it is possible to avoid this behavior, and concentrate them all in coherent regions. Clearly the disadvantage of the 'Specified GCP' method is that it is necessary for the user to do some extra work to obtain the full resolution coordinates of the required GCPs. This can be done, for example, using the QUICK LOOK GENERATION function with the grid in row,col coordinates.

Another important use of this feature is the registration of ascending and descending pass images. In this case, the various image features are observed from different angles and are therefore very difficult to correlate. One solution for this problem is to select as GCPs only high reflecting scatters (or similar regions). This can be done as explained before, using the QUICK LOOK GENERATION function with the grid in row,col coordinates.

The following example shows the format of the Specified GCP ASCII file, simply composed of coordinate pairs from the master image, defined in the row,col coordinate system and separated by paragraph marks:

```
100 200
244 772
844 902
1200 30
2309 4445
```

The Warp Function

Once the valid positions of the GCPs in the slave image(s) are known, a function is computed using a least squares method, which maps the GCPs in the slave image onto the GCPs in the master image. This function (usually know as the warp function) is used to perform the co-registration.

The warp function is a polynomial in the row,col coordinates with a degree defined by the parameter 'Transformation Degree'. This ranges from 1 (which corresponds to a simple transformation which includes only translation, rotation, scaling and skew of the slave image(s)) up to 3 (which is a rather complex transformation with no simple geometric explanation). The

forms of the warp functions for the four different degrees (1, 1.5, 2 or 3) are described in the 'Warp Evaluation' chapter of the Algorithm Specification Document [A3].

Important: As a simple rule, when the input images do not suffer from a high level of distortion try first a degree of 1 (or 1.5). This permits the evaluation of a smooth warp, which is enough for the majority of cases (and in particular should be sufficient for Tandem ERS data).

Larger values of the 'Transformation Degree' parameter should only be used when a very good co-registration accuracy is required. Polynomials of second or third degree can introduce large distortions in image regions containing only a few GCPs. In these cases it is advisable to use large numbers of GCPs, especially in areas that are of particular interest to the user.

GCP Residuals

The residuals (the errors introduced by the warping function) for each GCP are computed and written to a text file defined by the parameter 'Residual File Name'. This file also contains the warp coefficients generated by the co-registration process.

It is often very useful to check the information contained within the residual file to see if the co-registration process can be considered to have been successful. For example, the final figure shown in the file is the average RMS residual value (referred to in the file as Total RMS). This value can be used as an approximate figure of merit for the co-registration. The GCPs that are not used because they have coherence values falling below the coherence threshold are indicated with a "*" symbol next to the coherence value. GCPs that are excluded because they have exceeded the value of the 'Editing RMS' parameter, are indicated by a "-" symbol in place of the coherence value. The residual file can also provide a useful starting point for selecting GCPs to be used in a 'Specified GCPs' file to refine the co-registration.

Interpolation

After the warp function has been computed, the next step in the co-registration process is the interpolation of the slave image pixels onto the master image grid. The interpolating function can be selected using the parameter 'Interpolation Mode'.

The fastest (and least accurate) interpolators are the nearest neighbour, bilinear and constant shift, while the most accurate one is the cubic convolution. The most precise is the sinc, which uses a configurable kernel size to obtain the interpolated value. It is also possible to mix two different types of interpolation; one method for the row direction and another for the column direction (this facility may be useful depending on the distortions suffered by the SAR images, for example, for interferometric pairs, the distortion in the column direction is often very low compared to the row direction).

The following interpolators can be used:

- **Nearest Neighbour** takes the pixel nearest to the position determined by the warp function; has an intrinsic accuracy of ±0.5 pixel but is very fast.
- **Bilinear** uses three linear interpolations of the four pixel values around the position determined by the warp function. This interpolator does not work very well with complex data.
- **Cubic Convolution** uses five interpolations with a four-coefficient cubic convolution kernel applied to the sixteen pixels around the position determined by the warp function.

- **Sinc**, the best (and slowest) interpolator, uses a sinc kernel of a selectable size N, applied N+1 times to the N by N pixels around the position determined by the warp function. This is the interpolator that is used by default, if no other interpolator is specified by the user.
- **Constant Shift**, in which each image block that has to be interpolated is shifted by a unique value (determined by the interpolation grid) by means of a FFT operation.
- **Sinc along rows and Constant Shift along columns**
- **Cubic Convolution along rows and Constant Shift along columns**

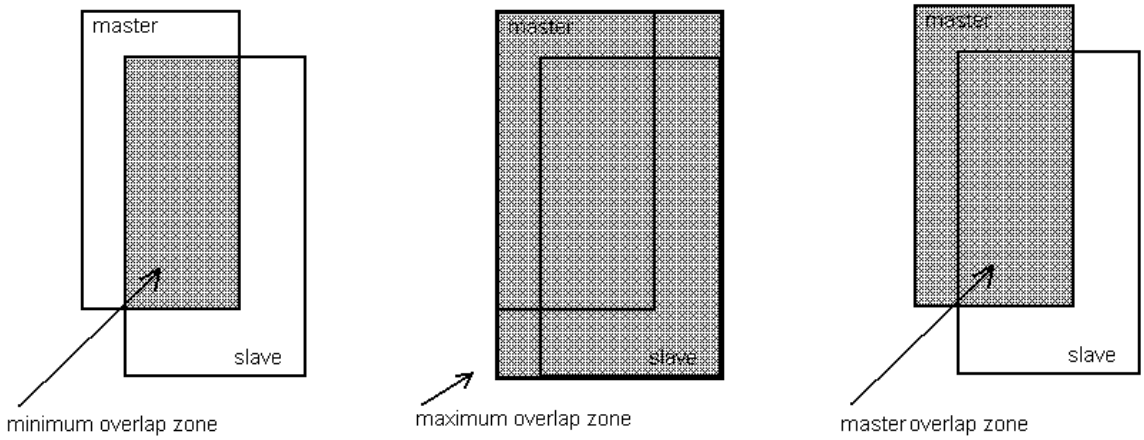
Important: For complex images, the bilinear and the cubic convolution interpolators do not work very well, because they change the azimuth spectra of the images causing unwanted effects on images with low coherence. Instead the constant shift (at least in the row direction) or the sinc interpolator is recommended, because these are the only interpolators that preserve the images' spectra.

Overlap Selection

The common zone in the master and slave images that shall be co-registered and written to the output files can be defined in a variety of ways, selected using the parameter 'Overlapping Mode'.

- **AOI**, in which the common portion to be processed is specified by the user. Any kind of AOI can be used except the example image mode.
- **Minimum Overlap**, where the common portion to be processed contains the pixels which are present both in the master and in all the slaves (this portion corresponds to the common overlap zone between all the images of the input stack).
- **Maximum Overlap**, where the full extents of all images are processed (pixels are present in at least one image, the portion corresponds to the maximum extents of the input stack).
- **Master Overlap**, where the common portion to be processed contains the pixels present in the master image.

Note that the overlapping area among the images is estimated at the beginning of the processing chain. If it is found to be less then the threshold established in the parameter 'Overlapping AoI Threshold', the program ends with an error.



Baseline Calculation

For real or complex images, the baseline between the two satellites is calculated using the orbit information and the warp function. The baseline is evaluated at the scene centre (both the normal and parallel components) in the same coordinate system that the orbit is given for each slave image. It is possible for the user to specify the name of the text file that will contain these values using the parameter ‘Baseline File Name’.

The following example shows the format of the baseline file for a single slave image:

```
### Slave number 1

 Baseline Cartesian Components in meters:

        X               Y               Z
 ---------------------------------------------
   -30.535631      -60.703657      19.806008

 Baseline Components in the Sat - Target Plane in meters:

           Normal            Parallel
 ---------------------------------------------
         62.978227         32.301387
```

Example "INI" files

The following “*.ini*” files are examples for the CO-REGISTRATION tool. Two cases are shown: the most basic, with the minimum set of parameters and a more complicated one, with customised configuration parameters.

```
[IMAGE CO-REGISTRATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Images = "slc_master.XTt", "slc_slave.XTt"
Output Images = "master", "slave"

[IMAGE CO-REGISTRATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Images = "pri_master.XTt", "pri_slave.XTt"
Output Images = "master", "slave"
Coordinate System = "ROWCOL"
Top Left Corner = 100,200
Bottom Right Corner = 1500,3000
GCPs Numbers = 15, 15
Image Coarse Reg = 'N'
Image Fine Reg = 'N'
Coarse Reg Window Sizes = 101, 101
Coarse Reg Interp Factors = 3.0, 3.0
Coherence Window Size = 7
Transformation Degree = 1
Overlapping Mode = "MIN"
Interpolation Mode = "SINC"
Baseline File Name = "basel.txt"
Residual File Name = "residual.txt"
```

Parameter Summary: Co-registration

Input Images

The name of the input images in internal format to be co-registered. The master image is the first image specified. A maximum of 10 images can be input. All images must be of the same type.

Example: `Input Images = "mas.XTt", "sla1.XTt", "sla2.XTt", "sla3.XTt"`

mandatory INPUT

BEST extension: ".??t", ".??s", ".??f", ".??c" where "??" indicates it is not important which module created the files, as long as the data type is correct.

AOI specification

see Appendix 4; no example image can be used. Note that the AOI specification prevents the use of the minimum, maximum and master overlap schemes.

optional parameter

Output Images

The names of the output images in internal format containing the co-registered images (an extension "*.CRf*" or "*.CRc*" is automatically added by the system).

Example: `Output Images = "masr", "sla1r", "sla2r", "sla3r"`

mandatory OUTPUT

BEST extension: ".CRf" or ".CRc"

Baseline File Name

The name of the text file which will contain the baseline information, evaluated for each slave image.

Example: `Baseline File Name = "basel_values.txt"`

optional OUTPUT (default is "*baseline.txt*")

Image Coarse Reg

Determines whether the 2nd, coarse registration step should be performed on the data.

Example: `Image Coarse Reg = "N"`

optional parameter (default is "Y")

GCPs Numbers

The number of GCPs that will be used in the co-registration, defined by the dimensions of a grid expressed in row,col. The cells will be generated at the intersections of the grid. This parameter is ignorred when the 'GCPs File Name' is defined.

Example: `GCPs Numbers = 7, 5`

optional parameter (default is "10, 10")

GCPs File Name

The name of a text file containing user selected GCPs, expressed in row,col full resolution coordinates on the master image, for use in the co-registration process. This parameter defines an alternative to the GCPs generated automatically if the parameter 'GCPs Numbers' is defined. If the parameter is not defined, GCPs are generated automatically according to the 'GCPs Numbers' parameter.

Example: `GCPs File Name = "GCPFile.dat"`

optional INPUT

Editing RMS

The threshold (in units of pixels) to exclude from the warp function those GCPs that produce high residual errors.
Example: `Editing RMS = 1.0`
optional parameter (default is “1.0”)

Coarse Reg Window Sizes

The size of the cells used for the coarse registration step, expressed in row.col. Each cell is centred on one of the GCP positions and the cross correlation is performed within it.
Example: `Coarse Reg Window Sizes = 51, 51`
optional parameter (default is “51, 51”)

Coarse Reg Interp Factors

The interpolation factors for the coarse registration step. The values are expressed as row,col in units of $pixel^{-1}$. This parameter defines the step size used in the cell cross correlation process. Higher values produce good accuracy at the expense of longer running times.
Example: `Coarse Reg Interp Factors = 3.5, 3.5`
optional parameter (default is “1.0, 1.0”)

Coarse Reg Tolerance

The coarse registration step is performed twice, using slightly different cell positions. If the final GCP positions from these two steps do not agree to within the limit set by this parameter, then these GCPs are not used in the remainder of the co-registration process.
Example: `Coarse Reg Tolerance = 1.0`
optional parameter (default is “1.1”)

Image Fine Reg

Determines whether the 3rd, fine registration step should be performed on the data. For complex data this will entail the maximisation of the complex coherence. For real data it will entail the refinement of the cross-correlation function.
Example: `Image Fine Reg = "N"`
optional parameter (default is “Y” for complex data and “N” for real data)

Fine Reg Window Sizes

The size of the cells used for the fine registration step, expressed in row.col. Each cell is centred on one of the GCP positions and the coherence maximisation or cross correlation refinement is performed within it.
Example: `Fine Reg Window Sizes = 51, 51`
optional parameter (default is “51, 51”)

Coherence Threshold

Threshold below which GCPs are excluded from the co-registration calculation.
Example: `Coherence threshold = 0.4`
optonal parameter

Coherence Window Size

The size of the square kernel used for the coherence evaluation in the fine registration step, expressed as the length of one side in pixels.
Example: `Coherence Window Size = 7`
optional parameter (default is “3”)

Coherence Func Tolerance

A stop criteria for the iterative searching during coherence maximization. When coherence changes fall below this tolerance, the search stops.
Example: `Coherence Func Tolerance = 1.e-6`
optional parameter (default is “1.e-6”)

Coherence Value Tolerance

A stop criteria for the iterative searching during coherence maximization. When the shifts (in units of pixels) made on the slave cell fall below this tolerance, the search stops.
Example: `Coherence Value Tolerance = 1.e- 3`
optional parameter (default is “1.e-3”)

Transformation Degree

The degree of the warp transformation polynomial:

- 1
- 1.5
- 2
- 3

Example: `Transformation Degree = 1.5`
optional parameter default is “1.5”)

Interp Window Sizes

The size in pixels of the processing image blocks into which the master image is subdivided. This affects the accuracy only when the ‘Interpolation Mode’ is set to “CONSTANT SHIFT” (in one or both directions); however it always affects the running time (it is faster to use large blocks).
Example: `Interp Window Sizes = 150, 150`
optional parameter (default is “512, 512”)

Interpolation Mode

The interpolation method:

- “NEAREST NEIGHBOUR”
- “BILINEAR”
- “CONSTANT SHIFT”
- “CUBIC CONVOLUTION”
- “SINC”
- “CONSTANT SHIFT CUBIC CONV”
- “CONSTANT SHIFT SINC”

Note that the cubic convolution does not work very well with complex data. It is strongly recommended that a constant shift or sinc interpolator is used for complex data.
Example: `Interpolation Mode = "CUBIC CONVOLUTION"`
optional parameter (default is “SINC”)

Interpolation Inverse Precision

The length of lookup tables used to speed up the sinc interpolation. High values give a good accuracy at the expense of an increased memory requirement. The accuracy that can be achieved is given by the reciprocal of this parameter, i.e. the default value (1000) gives an accuracy of 1/1000 of a pixel.
Example: `Interpolation Inverse Preci sion = 1000`
optional parameter (default is “1000”)

Sinc Width

The size of the sinc interpolation kernel, in units of pixels. This value affects the accuracy of the interpolation at the expense of the processing time.
Example: `Sinc Width = 7`
optional parameter (default is “7”)

Cubic Convolution Coefficient

A coefficient which modifies the behaviour of the cubic convolution interpolator; the IDL cubic interpolation uses a coefficient equal to -1 while ERDAS suggest +0.5.
Example: `Cubic Convolution Coefficient = -1.0`
optional parameter (default is “-1”)

Overlapping Mode

The type of overlapping scheme:

- “MIN”
- “MAX”
- “MASTER”

Example: `Overlapping Mode = "MASTER"`
optional parameter (default is “MASTER”)

Residual File Name

The name of the text file which will contain the warp coefficients and the residuals for each GCP.
Example: `Residual File Name = "residual"`
optional OUTPUT (default is “*residual.txt*”)

Overlapping AoI Threshold

This parameter is the minimum allowable common area between the master image and the slaves, expressed as a percentage of the master image area. The minimum value for which co-registration is still considered possible is assumed to be around 30%.
Example: `Overlapping AoI Threshold = 10`
optional parameter (default is “30”)

Coherence Generation

Description

The COHERENCE GENERATION tool computes the coherence image between two co-registered complex or real SAR products. The coherence is generated in a window of a user-defined size, which moves with a step size of 1 pixel across the images.

The input data must be complex or real, in the Toolbox internal format and with floating point pixels. Images cannot be input if they have been extracted directly from SAR products, in which case the pixel has an integer or complex integer format.

Complex data are processed by evaluating the modulus of the complex correlation coefficient; only the real correlation coefficient is considered for real images.

The coherence generation function produces an output image with the same size as the input couple. Note that there will be an ‘edge effect’ caused by the size of the moving window in which the coherence is calculated. This effect will cause a buffer of pixels (with a depth equal to half the window size) to be set to zero at the edges of the output image.

Example "INI" file

```
[COHERENCE IMAGE GENERATION]
Input Dir = "./"
Output Dir = "./"
Input Images = "slc_master.CRc", "slc_slave.CRc"
Output Image = "cohe"
Window Sizes = 7, 7
```

Parameter Summary: Coherence Generation

Input Images

The name of the input image couple in internal format from which the coherence is generated.
Example: `Input Images = "mas.CRc", "sla.CRc"`
mandatory INPUT
BEST extension: “.??f”, “.??c” where "??" indicates it is not important which module created the files, as long as the data type is correct.

AOI specification

see Appendix 4; neither example image nor polygonal specification can be used.
optional parameter (default is entire input image)

Window Sizes

The size of the moving window used to compute the coherence, expressed as row,col.
Example: `Window Sizes = 5, 5`
mandatory parameter

Output Image

The name of the output image in internal format containing the coherence information (an extension “*.CHf*” is automatically added by the system)
Example: `Output Image = "cohe"`
mandatory OUTPUT

BEST extension: “.CHf”

Footprint Registration

Description

The FOOTPRINT REGISTRATION tool generates a standard TIFF format quick look version of a master image, superimposed with the outlines of one or many slave images. This offers a fast method of assessing the suitability of images in a dataset for co-registration purposes.

The tool uses an approximated method based on the corner localisation information from the Geolocation Annotation in the product headers to locate the slave image outlines relative to the master.

No AOI is permitted for this function

Example "INI" file

```
[FOOTPRINT REGISTRATION]
Input Dir = "./"
Output Dir = "./"
Input Master Image = "asar_aps.XTt"
Input Images = "slave_1.XTt", "slave_2.XTt"
Output Image Size = 1485 ,490
Output Image = "footprint"
Delete Input Image = "N
```

Parameter Summary: Footprint Registration

Input Master Image

The name of the input master image over which the slave image(s) must be co-registered.

Example: `Input Master Image = "asar_aps.XTt"`

mandatory INPUT

BEST extension: ".??i", ".??s", ".??t", ".??f" or ".??c" where "??" indicates that any BEST module could have produced these files.

Input Images

The name of the input slave image(s) to be co-registered and superimposed.

Example: `Input Images = "slave_1.XTt", "slave_2.XTt"`

mandatory INPUT

BEST extension: ".??i", ".??s", ".??t", ".??f", ".??c" where "??" indicates that any BEST module could have produced these files.

Output Image Size

The size of the output TIFF image that contains the master image superimposed with the slave's outlines, expressed in row,col.

Example: `Output Image Size = 1485 ,490`

To maintain the aspect ratio of the input data, set one of the values to "0". This causes the system to compute a size that maintains square pixels.

To generate an output image with 1400 rows and square pixels use:

```
Output Image Size = 1400, 0
```

To generate an output image with 500 columns and square pixels use:

```
Output Image Size = 0, 500
```

mandatory parameter (default is "1000, 0")

Output Image

The name of the output file that contains the footprint co-registration image (the extension "*.tif*" is automatically added by the system)

Example: `Output Image = "footprint"`

mandatory OUTPUT

BEST extension: ".tif"

Image Geo-correction

Description

The GEO-CORRECTION tool performs a geocoding process to georeference input ASAR **Medium Resolution** (i.e. ASA_IMM_1, ASA_WSM_1 or ASA_APM_1) real images. It uses the Geolocation Annotation in the product header to reproject the data to a flat earth ellipsoid (no kind of terrain relief is considered for such operation). The output image is hence distorted so that its vertical and horizontal axes are aligned to the North and East axes of the selected cartographic projection (UTM or UPS).

The Geocorrection is based on the following steps:

- creation of a regular grid in the selected cartographic reference system, having a spacing between the nodes equal to the input pixel and line spacing.
- transformation of the grid nodes from cartographic into input image coordinates (row,col).
- interpolation of the input image to generate the output image, using the previously generated grid.

The new annotated information (new corner localisation, the correspondence between lat,lon and row,col, etc.) is computed and updated in the output file.

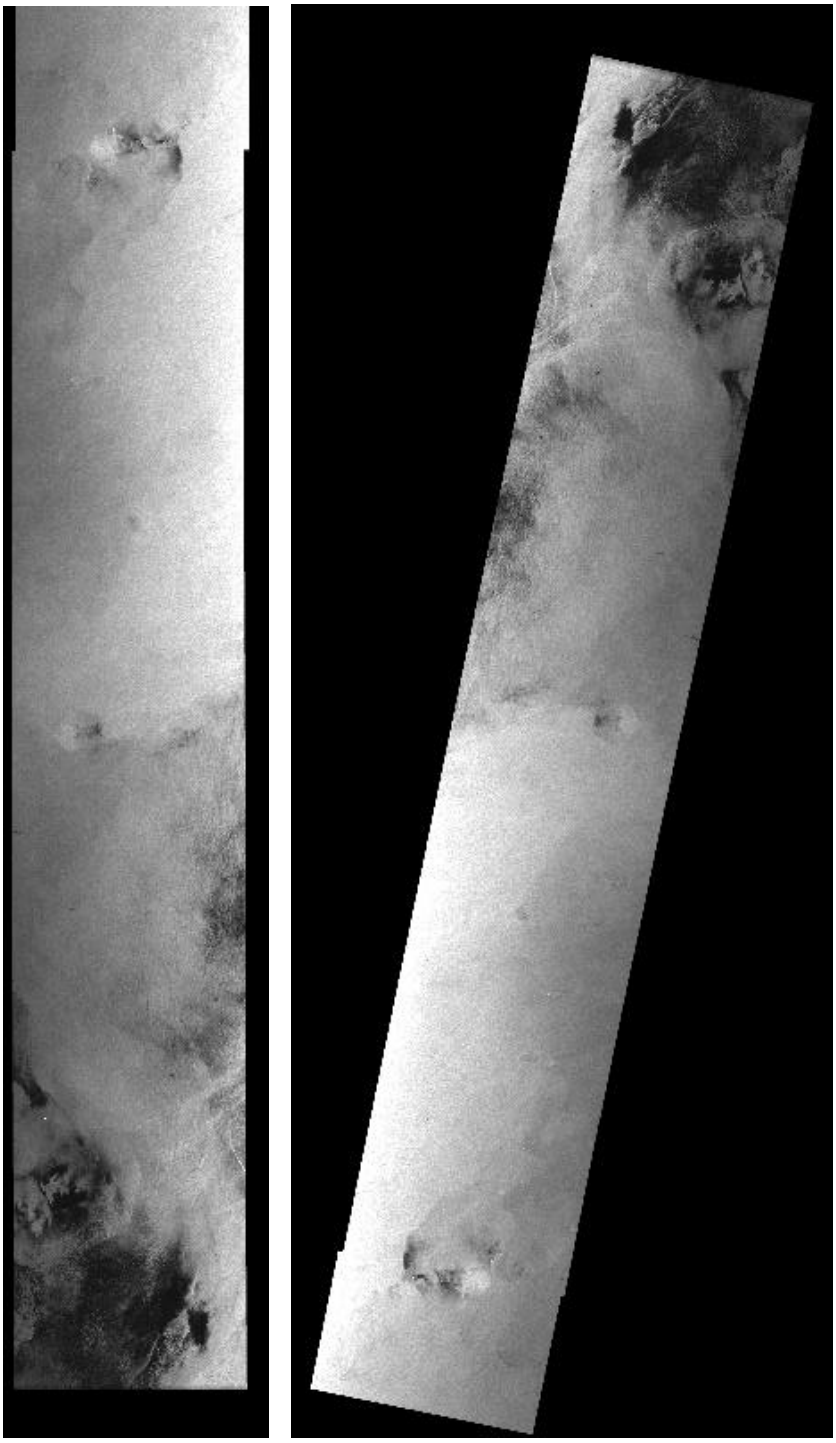
The method of interpolation is selected using the ‘Interpolation Mode’ parameter. The fastest (and least accurate) method is the bilinear interpolator; the most accurate is cubic convolution. The sinc interpolator is most precise.

The following interpolators may be used:

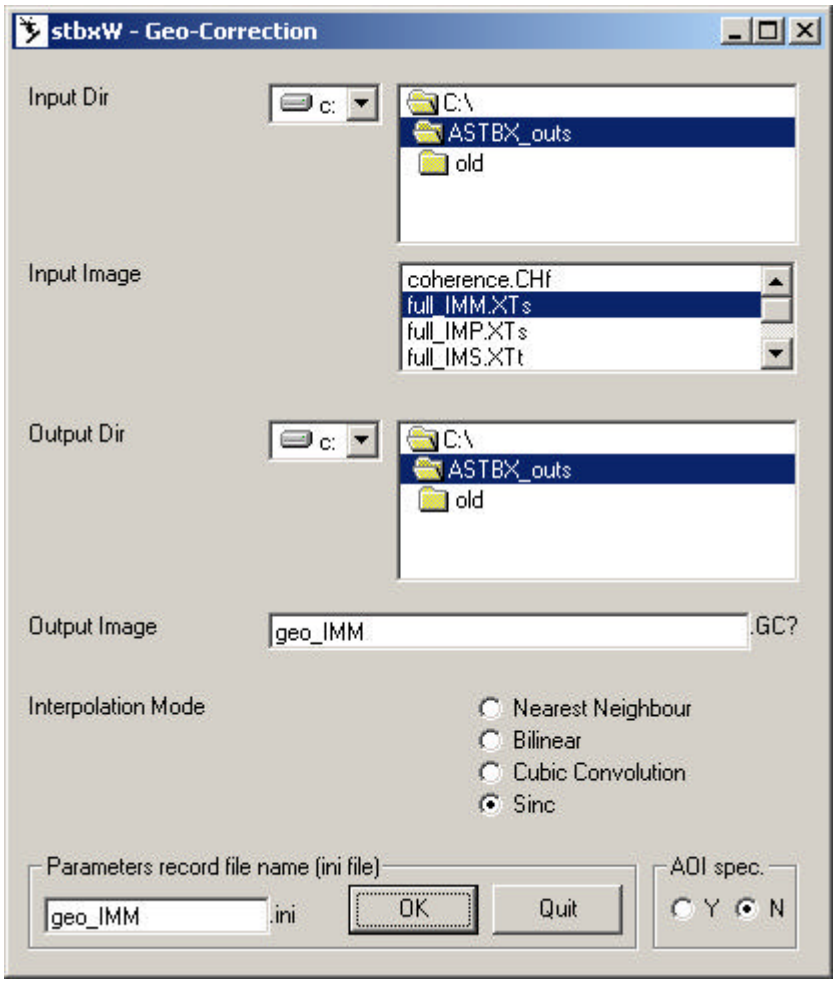
- **Bilinear** uses three linear interpolations of the four pixel values around the position determined by the transformation function. This interpolator does not work very well with complex data.
- **Cubic Convolution** uses five interpolations with a four-coefficient cubic convolution kernel applied to the sixteen pixels around the position determined by the transformation function. This is the interpolator that is used by default, if no other interpolator is specified by the user. However, it should be noted that this interpolator does not work very well with complex data; in such cases the sinc interpolator is recommended.
- **Sinc**, the best (and slowest) interpolator, uses a sinc kernel of size N, applied N+1 times to the N by N pixels around the position determined by the transformation function.

The AOIs may be defined by the rectangular (with corners expressed in row,col or in lat,lon coordinates) or polygonal (in this case the surrounding rectangular AOI is used) methods.

The figures below show quick look images of an ASA_APM_1P product before and after geo-correction.



HMI



Typical HMI settings for an ASA_IMM_1P product

Typical Processing Chain

HEADER ANALYSIS ⇒ FULL RESOLUTION ⇒ IMAGE GEO-CORRECTION

Example "INI" file

```
[IMAGE GEO-CORRECTION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "full_IMM.XTs"
Output Image = "geo_IMM"
Interpolation Mode = "SINC"
```

Parameter Summary: Image Geo-correction

Input Image

The name of the input ASAR Medium Resolution image in internal format
Example: `Input Image = "full_IMM.XTs"`
mandatory INPUT
BEST extension: ".??s" or ".??f" where "??" indicates that any BEST module could have produced this file.

Output Image

The name to be given to the internal format image which will contain the geo-corrected image (an extension "*.GR?*" is automatically added by the system)

Example: `Output Image = "geo_IMM"`

mandatory OUTPUT

BEST extension: ".GRf"

AOI specification

see Appendix 4; for polygonal AOIs the surrounding rectangular AOI is used

optional parameter (default is entire input image)

Interpolation Mode

The method of interpolation:

- BILINEAR
- CUBIC CONVOLUTION
- SINC

Note that the cubic convolution does not work very well with complex data. It is strongly recommended that a constant shift or sinc interpolator is used for complex data.

Example: `Interpolation Mode = "SINC"`

optional parameter (default is "CUBIC CONVOLUTION")

Amplitude-Coherence Multi-layer Composite

Description

This AMPLITUDE-COHERENCE MULTI-LAYER COMPOSITE function generates an RGB colour image with three data layers obtained from an interferometric co-registered couple. The Red channel always contains the coherence of the interferometric couple. There are two options for the contents of the Green and Blue channels, giving the possibility for two different output colour composite products:

CAD:
Red - coherence
Green - average of the modulus backscatter images
Blue - difference between the modulus backscatter images

CMS:
Red - coherence
Green - modulus backscatter of the master image
Blue - modulus backscatter of the slave image

The layers are rescaled from 16-bit to 8-bit according to the following default methodology:

- **Coherence image:** a linear scaling between 0 and 0.9.
- **Modulus backscatter images:** a logarithmic scaling between -22 dB and 3.5 dB.
- **Modulus backscattering difference image:** stretched in such a way that the ratio $\left|10\log_{10}\left(\frac{I_2}{I_1}\right)\right|$ is scaled between 1.0 and 6.0 (where I_2 and I_1 are the intensity of the 2nd and 1st co-registered images respectively).

The user may alter the methodology by setting parameters in the "*.ini*" file. Alternatively, an external look up table may be applied afterwards as an 8-bit to 8-bit conversion (8 bits per layer).

No AOI is permitted in this function.

Example "INI" file

```
[AMPLITUDE-COHERENCE MULTI-LAYER COMPOSITE]
Input Dir = "./"
Output Dir = "./"
Coherence Image = "cohe.CHf"
Co-registred Images = "slc_master.CRc", " slc_slave.CRc"
Output Image = "mlayer"
Multi-layer Mode Flag = "CAD"
Coherence Upper Threshold = 0.75
Image Lower Threshold = -18
Image Upper Threshold = 2.5
DiffImage Lower Threshold = 2
DiffImage Upper Threshold = 5
```

Parameter Summary: Amplitude-Coherence Multi-layer Composite

Coherence Image

The name of the input coherence image
Example: `Coherence Image = "cohe.CHf"`
mandatory INPUT
BEST extension: ".CHf"

Co-registred Images
The name of the input interferometric co-registered image couple
Example: `Co-registred Images = "slc_master.CRc", "slc_slave.CRc"`
mandatory INPUT
BEST extension: ".CRc" or ".CRf"

Output Image
The name of the output true colour RGB multi-layer image in internal format
Example: `Output Image = "mlayer"`
mandatory OUTPUT
BEST extension: ".tif"

Multi-layer Mode Flag
The type of output colour composite image to generate:
- "CAD"
- "CMS"

Example: `Multi-layer Mode Flag = "CAD"`
mandatory parameter

Coherence Upper Threshold
The upper limit for the linear scaling of the coherence real image (stretching from 16-bit to 8-bit).
Example: `Coherence Upper Threshold = 0.75`
optional parameter (default is "0.9")

Image Lower Threshold
The lower limit in dB for logarithmic scaling of the average modulus backscatter computed from the two co-registered images.
Example: `Image Lower Threshold = -18`
optional parameter (default is "-22")

Image Upper Threshold
The upper limit in dB for logarithmic scaling of the average modulus backscatter computed from the two co-registered images.
Example: `Image Upper Threshold = 2.5`
optional parameter (default is "3.5")

DiffImage Lower Threshold
The lower limit for linear scaling of the ratio (in dB) of the two co-registered modulus backscatter images.
Example: `DiffImage Lower Threshold = 2`
optional parameter (default is "1")

DiffImage Upper Threshold
The upper limit for linear scaling of the ratio (in dB) of the two co-registered modulus backscatter images.
Example: `DiffImage Upper Threshold = 5`

optional parameter (default is “6”)

User LUT

The name of an ASCII file containing the lookup table used for further stretching of the 8-bit layers.

Example: `User LUT = "lut.dat"`

optional parameter

13. Speckle Filter

This chapter documents the following tools:

1. Speckle Filter
Removes speckle noise from real intensity images using the ‘Gamma MAP’ algorithm.

Speckle Filter

Description

The speckle filter tool removes speckle noise from intensity images using the ‘Gamma MAP’ algorithm. This is provided in the SAR Toolbox because removing the speckle noise from a SAR image is an important step towards producing a meaningful backscattering coefficient image. The speckle filter tool operates on real intensity images in the internal SAR Toolbox format.

If the input data is from a PRI image, this will initially be in amplitude form and will therefore first need to be converted to an intensity image (i.e. using the Amplitude to Power function).

The various parameter values shown in the following “Speckle Filter Summary Table” are those recommended for ERS ESA PRI products.

Or if the data is from a SLC image, the data will first need to be converted to a real form (i.e. Modulus extraction) followed by a conversion to intensity (i.e. the Amplitude to Power function).
Of course, if required, the output from the Speckle Filter function may be converted from intensity to amplitude form by using the Power to Amplitude function.

The Speckle Filter makes use of a range of different masks that are applied in a moving window that covers the input image. For each position of the window the masks are used to determine the structure of the dominate feature within the window. Once this structure has been discovered, the filter may either; preserve the pixel value at the centre of the window, replace it with the mean of the window values (in the case for homogeneous regions, or use the GAMMA - GAMMA MAP filter to evaluate the centre pixel value.

Further details about the masks used in the Speckle Filter algorithms are given after the Speckle Filter Summary Table.

Further characteristics of the speckle filter tool are:

- it uses only the rectangular AOI specification with corners expressed in row, col system
- it uses (for the standard masks) odd filter sizes ranging from 3 by 3 to 31 by 31 with a corresponding number of directions from 4 to 60 (the upper kernel size limitation is only imposed by the automatic computation of the thresholds; if these values are inserted from the user a greater kernel can be used)
- used predefined masks evaluating in an automatic way the edge and line thresholds
- can also apply user defined masks, once the edge and line thresholds are specified by the user
- updates in the output image, the following annotations:
  - the new values of the coordinates in lat, lon of the four corners and of the center point
  - the new entry in the processing history
- the speckle filtering function is interfaced with ERMAPPER using the “filter” feature

Example "INI" files

The following “.INI” file is an example for the speckle filtering of a PRI image portion with standard masks:

```
[SPECKLE FILTER]
Input Dir = "./"
Output Dir = "./"
Input Image = "t1_priimage.APf"
Top Left Corner = 50, 50
Bottom Right Corner = 500, 500
Window Sizes = 11, 11
PFA = 10.0
Scatter Threshold = 0.57
Output Image = "specklefiltered_img"
```

The following file does the same operation with user mask

```
[SPECKLE FILTER]
Input Dir = "./"
Output Dir = "./"
Input Image = "input.APf"
Top Left Corner = 50, 50
Bottom Right Corner = 500, 500
Window Sizes = 11, 11
PFA = 10.0
Scatter Threshold = 0.57
Mask File = "user_mask.ker"
Number of Look = 3.0
Edge Threshold = 0.82
Line Threshold = 0.87
Output Image = "specklefiltered_img"
```

Speckle Filter Summary Table

Parameter	Description Example	Comment
Input Image	the name of the real input image in internal format, containing intensity (i.e. power) data; this image can come from a PRI or a SLC data but in the latter case a modulus extraction shall be performed before the power conversion Example: Input Image = "t1_priimage.APf"	mandatory INPUT BEST extension: APf
AOI specification	only the Rectangular AOI specifications are permitted (see Appendix 4)	optional parameter; if not present, the entire input images are assumed
Window Sizes	the size of the speckle filter in pixel units (row,col ordering) For example, Window Sizes = 11, 11	mandatory parameter
PFA	the Probability of False Alarm in the vari ous regions recognition, expressed in per centage units (the lower is this value, less filtered is the output image, due to the fact that we are recognizing the various regions under a very strict criterion) For example, PFA = 10.0	mandatory parameter
Scatter Threshold	the threshold used for the scatter detection (higher is this value, less preserved are the scatters in the output image, due to the fact that we are recognizing the recognizing regions under a very strict criterion) For example, Scatter Threshold = 0.57	mandatory parameter
Output Image	the name of the output image in internal format containing the speckle filtered intensity image (an extension “SFf” is automatically added by the system) Output Image = "specklefiltered_img"	mandatory OUTPUT BEST extension: SFf
Mask File	the filename containing filtering masks defined by the user; the user masks are used instead of the standard when this parameter is present Mask File = "usermask.ker"	mandatory parameter for the filtering with user masks
Number of Look	the number of looks of the input image; when this parameter is absent from the "INI" file, then the value stored in the image annotations is used; due to the change of the image statistics after a image filtering operation, the output image has this value set to 0 in the annotations, thus	optional parameter for images directly coming from the extraction tool mandatory parameter for input images which have been previously filtered

	requiring the user to set a right value (e.g. obtained from measures on the filtered image) For example (for PRI) images, Number of Look = 3.0	
Edge Threshold	the value of the threshold for the ratio detector applied on the edge regions of user masks; this value should be computed using the relation showed in Speckle Filter chapter of the document; for standard masks this value is automatically com puted by the system but can be however overridden with this "INI" line For example, Edge Threshold = 0.82	optional parameter for standard masks mandatory parameter for user masks
Line Threshold	the value of the threshold for the ratio detector applied on the edge regions of user masks; this value should be computed using the relation showed in Speckle Filter chapter of the document; for standard masks this value is automatically com puted by the system but can be however overridden with this "INI" line For example, Line Threshold = 0.87	optional parameter for standard masks mandatory parameter for user masks

The filtering masks are contained in ASCII files and their structure is governed by the following rules:

all the masks for the various regions (edge, line, scatter) filtering are kept in a unique ASCIII file
all the masks shall have the same size, which defines the filter size
each row shall be terminated with a newline (return) character
the first row of each mask shall indicate
the mask type, chosen between “edgeline” and “scatter”
the mask identification, as a progressive number between 1 and the number of possible masks of a given type
the symbols used for the pixel description shall be separated by one space character
the symbol used to describe a left edge region is the “1”
the symbol used to describe a right edge region is the “4”
the symbol used to describe a line region is the “.”
the symbol used to describe a left buffer region is the “2”
the symbol used to describe a right buffer region is the “3”

The following listing shows the standard mask file of size 11:

```
edgeline 1
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
edgeline 2
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
3 3 4 4 4 4 4 4 4 4 4
. . 3 3 4 4 4 4 4 4 4
2 2 . . 3 3 3 4 4 4 4
1 1 2 2 . . . 3 3 4 4
1 1 1 1 2 2 2 . . 3 3
1 1 1 1 1 1 1 2 2 . .
1 1 1 1 1 1 1 1 1 2 2
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
edgeline 3
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 2 2
1 1 1 1 1 1 1 1 2 2 . .
1 1 1 1 2 2 2 . . 3 3
1 1 2 2 . . . 3 3 4 4
2 2 . . 3 3 3 4 4 4 4
. . 3 3 4 4 4 4 4 4 4
3 3 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
edgeline 4
. 3 4 4 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
1 2 . 3 4 4 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 1 1 2 . 3 4
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 1 1 2 .
edgeline 5
```

1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2
. . . . . .
3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
4 4 4 4 4 4 4 4 4 4 4
edgeline 6
1 1 1 1 1 1 1 1 1 2 .
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 2 . 3 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 2 . 3 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
. 3 4 4 4 4 4 4 4 4 4
edgeline 7
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
edgeline 8
1 1 2 . 3 4 4 4 4 4 4
1 1 2 . 3 4 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 2 . 3 4 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 2 . 3 4 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 2 . 3 4 4 4
1 1 1 1 1 1 2 . 3 4 4
1 1 1 1 1 1 2 . 3 4 4

```
scatter 1
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . X . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
```

The user masks can be built in an analogous way. The most important thing to remember when building user mask are:

for user masks, the edge and line thresholds cannot be computed by the system (refer to the Speckle Filter chapter of the document [A3] for details) and shall be given by the user (the mathematical computations are however quite complex and also use empirical formulas); in the Speckle Filter chapter of the document [A3] some values are given for certain filter sizes and PFA values

the various regions (e.g. the line or the edge) shall always be centred on the central mask pixel

The user masks can be useful if it is required that the speckle filter recognize special image features, like the curved line in the following mask:

```
edgeline 1
2 . 3 4 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
2 . 3 4 4 4 4 4 4 4 4
2 . 3 3 4 4 4 4 4 4 4
2 2 . . 3 3 3 4 4 4 4
1 1 2 2 . . . 3 3 4 4
1 1 1 1 2 2 2 . . 3 3
1 1 1 1 1 1 1 2 2 . 3
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 1 2 . 3
1 1 1 1 1 1 1 1 2 . 3
```

14. Calibration

This chapter documents the following tools:

For ERS data:

1. Backscattering Image Generation
Converts a power image into a backscatter image.

2. ADC Compensation
Corrects a power image for the ADC saturation phenomenon in ERS SAR products (prior to BACKSCATTERING IMAGE GENERATION).

3. Gamma Image Generation
Converts a backscatter image (i.e. output from BACKSCATTERING IMAGE GENERATION) into a Gamma image by dividing by the cosine of the incidence angle.

For ASAR data:

4. Backscattering Image Generation
Converts a power image into a backscatter image.

5. Retro-calibration
Removes an annotated antenna pattern and replaces it with another one.

6. Rough-range Calibration
Corrects ASAR Wide Swath and Global Monitoring Mode images for the effect of incidence angle variation from near to far range.

7. Enhancement Swath
Corrects ASAR Wide Swath and Global Monitoring Mode products affected by intensity discontinuities between sub-swaths

Backscattering Image Generation (ERS)

Description

The BACKSCATTERING IMAGE GENERATION tool is used to convert a power image into an image of backscattering intensity. The output image may have either a linear or dB scale.

The following radiometric effects that could give a poor quality backscattering image can be corrected with this tool:

- antenna pattern
- range spreading losses
- replica power variation
- ADC saturation effect

The first two affect only SLC data, which comes from the PAF in an uncorrected form. The last two may affect the radiometry of any product.

The antenna pattern correction (application or removal) uses as input an Antenna Pattern File. Nominal files for ERS1 and ERS2 are provided with the BEST software release (located in the '*./cfg*' directory) although others can be created, if required, using the SUPPORT DATA INGESTION tool.

The reference replica power values used (in linear scale) are:

205229.0 for ERS-1
156000.0 for ERS-2

The reference chirp average density values used are:

267200 for ERS-1
201060 for ERS-2

When the ADC saturation correction is required, an ADC correction image shall also be provided as input. This correction image should be previously generated using the ADC CORRECTION image generation tool described on the following page. The BACKSCATTERING IMAGE GENERATION tool needs a value for the ADC saturation correction for every pixel in the image, so the ADC correction image must be computed using the same image portion used here (or a larger dataset from which a subset used here was taken). An error message is issued when this condition is not respected.

Example "INI" files

The following "*.ini*" file is the simplest example for backscattering image generation from a PRI power image, without any correction for the replica power variation or ADC saturation:

```
[IMAGE BACKSCATTERING]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "pri.APf"
Calibration Constant Correction = "APPLY"
Output Image Scale = "DB"
Output Image = "pri_s0"
```

The following "*.ini*" file is an example of fully compensated backscattering image generation from a PRI power image processed at a PAF that annotates the replica power value (in the PRI the antenna pattern and the range spreading loss are already compensated during the SAR processing):

```
[IMAGE BACKSCATTERING]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "pri.APf"
Replica Power Correction = "APPLY"
Reference Replica Power = 205229.0, 156000.0
ADC Saturation Correction = "APPLY"
ADC Saturation Correction File = "pri_adc.ADf"
Calibration Constant Correction = "APPLY"
Output Image Scale = "DB"
Output Image = "pri_s0"
```

The following "*.ini*" file is an example of fully compensated backscattering image generation from a SLC power image processed at a PAF that annotates the chirp density value (in the SLC the antenna pattern and the range spreading loss are not corrected during the SAR processing and shall be compensated here):

```
[IMAGE BACKSCATTERING]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "slc.APf"
Antenna Pattern Correction = "APPLY"
Range Spreading Loss Correction = "APPLY"
Replica Power Correction = "APPLY"
Reference Chirp Average Density = 267200, 201060
ADC Saturation Correction = "APPLY"
ADC Saturation Correction File = "slc_adc.ADf"
Calibration Constant Correction = "APPLY"
Output Image Scale = "DB"
Output Image = "slc_s0"
```

Typical Processing Chain

In the case when ADC saturation correction is required, the sequence of processing steps could be the following:

PORTION EXTRACTION ⇒ AMPLITUDE TO POWER ⇒ ADC COMPENSATION GENERATION ⇒ IMAGE BACKSCATTERING

Parameter Summary: Backscattering Image Generation (ERS)

Input Image

The name of the input image in internal format containing intensity (power) data, from which the backscattering image will be generated.
Example: `Input Image = "pri.APf"`
mandatory INPUT
BEST extension: ".APf"

AOI specification

See Appendix 4; the example image mode is not permitted.

optional parameter (default is entire input image)

Antenna Pattern Correction

Determines whether the antenna pattern compensation factor shall be applied (“APPLY”) or removed (“REMOVE”) from the image; if the parameter is omitted, this correction is not considered at all (neither applied nor removed).
Example: `Antenna Pattern Correction = "APPLY"`
optional parameter (default is no correction)

Antenna Pattern File

The name of the internal format file containing the antenna pattern.
Example: `Antenna Pattern File = “ers2.SDf”`
mandatory INPUT if ‘Antenna Pattern Correction’ is set to “APPLY”
BEST extension: “.SDf”

Range Spreading Loss Correction

Determines whether the range spreading loss compensation factor shall be applied (“APPLY”) or removed (“REMOVE”) from the image; if the parameter is omitted, this correction is not considered at all (neither applied nor removed).
Example: `Range Spreading Loss Correction = "APPLY"`
optional parameter (default is no correction)

Replica Power Correction

Determines whether the replica power variation compensation factor shall be applied (“APPLY”) or removed (“REMOVE”) from the image; if the parameter is omitted, this correction is not considered at all (neither applied nor removed).
Example: `Replica Power Correction = "APPLY"`
optional parameter (default is no correction)

ADC Saturation Correction

Determines whether the ADC saturation compensation factor shall be applied (“APPLY”); if the parameter is omitted, no correction is considered (this correction cannot be removed from the image).
Example: `ADC Saturation Correction = "APPLY"`
optional parameter (default is no correction)

ADC Saturation Correction File

The name of the internal format file containing the ADC saturation correction image (generated using the ADC CORRECTION image generation tool).
Example: `ADC Saturation Correction File = "adc.ADf"`
mandatory INPUT if ‘ADC Saturation Correction’ is set to “APPLY”
BEST extension: “.ADf”

Calibration Constant Correction

Determines whether backscattering values shall be computed from an input power image (“APPLY”) or if the inverse transformation, from backscattering image to original power image, shall be applied (“REMOVE”); if the parameter is omitted, no transformation is performed (neither in one direction nor in the other). Omitting the parameter allows one or more correction factors to be applied to the input image without altering the image type.
Example: `Calibration Constant Correction = "APPLY"`
optional parameter (default is no change to image type)

Calibration Constant

A user-defined value for the calibration constant; if missing, the value contained in the image annotations is used.

Example: `Calibration Constant = 1000000.0`

optional parameter

Output Image Scale

The scale of the output backscatter image:

- "LINEAR"
- "DB"

Do not use the dB scale if a further step of averaging is foreseen.

Example: `Output Image Scale = "DB"`

optional parameter (default is "LINEAR")

Output Image

The name of the output image in internal format containing the backscatter data (an extension "*.BSf*" is automatically added by the system).

Example: `Output Image = "backscatt"`

mandatory OUTPUT

BEST extension: ".BSf"

ADC Compensation (ERS)

Description

The ADC CORRECTION image generation tool computes the ADC compensation image that is required by the BACKSCATTERING IMAGE GENERATION tool to correct for the ADC saturation effect. This effect (present in all ERS images but particularly those from ERS-1) can alter the derived backscattering values on high reflectivity zones.

The ADC image generation is based on two filters with averaging kernels. The first, called RMS averaging, is used to reduce the computational load. The second, called smoothing, uses a window size dependent on the length of the functions used during the original SAR processing (which vary between SLC and PRI products) as follows:

Smoothing window size for PRI products = 400 rows, 1200 columns
Smoothing window size for SLC products = 630 rows, 1280 columns

During the ADC compensation image generation, some radiometric corrections previously applied to the input image have to be removed. Therefore, certain related parameters (used during the backscattering image generation to apply the radiometric correction) are required as input. It is important that these parameters are unchanged between the ADC compensation image generation and the backscattering image generation. As an example, if a user wants to specify a customised calibration constant during the backscattering image generation, the same value must be specified here.

The ADC CORRECTION image generation tool uses as input an ADC lookup table in internal format. Nominal files for ERS1 and ERS2 are provided with the BEST software release (located in the '*./cfg*' directory) although others can be created, if required, using the SUPPORT DATA INGESTION tool.

The reference replica power values used (in linear scale) are:

205229.0 for ERS-1
156000.0 for ERS-2

The reference chirp average density values used are:

267200 for ERS-1
201060 for ERS-2

The ADC correction image must be evaluated on the same power image (or the image from which a portion has been extracted) that will be input to the BACKSCATTERING IMAGE GENERATION tool. No Area of Interest (AOI) parameters can be used with the ADC CORRECTION tool, but it can work on image portions.

Example ".INI" file

```
[ADC COMPENSATION GENERATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "pri.APf"
```

```
RMS Window Size = 8, 8
PRI Smoothing Window Size = 400, 1200
SLC Smoothing Window Size = 630, 1280
Reference Replica Power = 205229.0, 156000.0
Reference Chirp Average Density = 267200, 201060
Output Image = "pri_adc"
```

Parameter Summary: ADC Compensation

Input Image

The name of the input image in internal format containing intensity (power) data, from which the ADC compensation image will be generated. The image shall be the same or contain the image used for the subsequent backscattering image generation.

Example: `Input Image = "pri.APf"`

mandatory INPUT

BEST extension: ".APf"

Calibration Constant

A user-defined value for the calibration constant; if missing, the value contained in the image annotations is used. Where used, this parameter must be the same as that specified in the subsequent backscattering image generation.

Example: `Calibration Constant = 9500000.0`

mandatory parameter IF the calibration constant is specified in the subsequent backscattering image generation

RMS Window Size

The size of the RMS averaging filter (used to reduce the input image for computational efficiency) in row,col.

Example: `RMS Window Size = 8, 8`

mandatory parameter

Output Image

The name of the output image in internal format containing the backscatter data (the extension "*.ADf*" is automatically added by the system).

Example: `Output Image = "pri_adc"`

mandatory OUTPUT

BEST extension: ".ADf"

Gamma Image Generation (ERS)

Description

The GAMMA IMAGE GENERATION tool converts a backscatter image (i.e. output from the BACKSCATTERING IMAGE GENERATION tool) into a Gamma image. This is achieved by dividing the backscatter image by the cosine of the satellite incidence angle. For certain terrain types the application of this function will make the backscatter image independent of the satellite incidence angle.

The area of interest (AOI) of the input image can be specified in any way (except the example image mode) and the output image may have either a linear or dB scale.

Example ".INI" file

```
[IMAGE GAMMA]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "sigma_0.BSf"
Output Image Scale = "DB"
Output Image = "gamma"
```

Parameter Summary: Gamma Image Generation

Input Image

The name of the input image in internal format containing backscatter data.
Example: `Input Image = "sigma_0.APf"`
mandatory INPUT
BEST extension: ".BSf"

AOI specification

See Appendix 4; the example image mode is not permitted.
optional parameter (default is entire input image)

Output Image Scale

The scale of the output backscatter image:

- "LINEAR"
- "DB"

Do not use the dB scale if a further step of averaging is foreseen.
Example: `Output Image Scale = "DB"`
optional parameter (default is "LINEAR")

Output Image

The name of the output image in internal format containing the backscatter data (the extension "*.GAf*" is automatically added by the system).
Example: `Output Image = "gamma"`
mandatory OUTPUT
BEST extension: ".GAf"

Backscattering Image Generation (ASAR)

Description

The BACKSCATTERING IMAGE GENERATION tool converts a power image into a backscatter image. The following radiometric effects are corrected:

- incidence angle
- absolute calibration constant
- antenna pattern
- range spreading loss

Only the first two need be applied to detected ground range products. For slant-range complex products, the last two must also be compensated. This differentiation is automatically performed by the system.

The output image may have either a linear or dB scale.

Example ".INI" file

```
[IMAGE BACKSCATTERING]
Input Dir = "G:\backscattering\power\"
Output Dir = "G:\backscattering\calib-prod\power\"
Input Image = "power_IMS2166.APf"
Output Image Scale = "LINEAR"
Output Image = "bs_pow_IMP4399"
Sensor Id = "ENVI"
```

Parameter Summary: Backscattering Image Generation (ASAR)

Input Image

The name of the input power image in internal format.
Example: `Input Image = "power_IMS2166.APf"`
mandatory INPUT
BEST extensions: ".APf", ".XTf", ".IT?", ".GT?", ".OV?", ".UNf", ".DBf", ".OP?", ".SGf", ".SGc", ".FI?", ".CR?", where "?" indicates that it is not important what format the data is in.

Output Image Scale

The scale of the output backscatter image:
- "LINEAR"
- "DB"

Do not use the dB scale if a further step of averaging is foreseen.
Example: `Output Image Scale = "LINEAR"`
mandatory INPUT

Calibration Constant

A user-defined value for the calibration constant; if missing, the value contained in the auxiliary file is used.
Example: `Calibration Constant = 34994.516`
optional parameter

Sensor Id

The platform from which the input data was acquired:

- “ERS”
- “ENVI” (Envisat)

Example: `Sensor Id = "ENVI"`
optional parameter

Output Image

The name of the output image in internal format containing the backscatter data (an extension “*.BSf*” is automatically added by the system).
Example: `Output Image = "bs_pow_IMP4399"`
mandatory OUTPUT
BEST extension: “.BSf”

Image Retro-calibration (ASAR)

Description

The IMAGE RETROCALIBRATION tool is used to remove an annotated antenna pattern and replace it with another one.

The function is useful in cases when routine instrument calibration exercises reveal that the antenna pattern for recently acquired data could be better estimated with a new pattern. In such cases, ESA generates a new External Calibration File (XCA) which may supersede one used to process a certain product originally.

The XCA file used to process the product is annotated in the SPH. The most recent XCA files are available for download from the ESA website (http://earth.esa.int/services/auxiliary_data/asar/). If a product was acquired **before** the creation date of the latest applicable XCA file, then it could have been processed with an older XCA file. The calibration would be more accurate if the data were to be retro-calibrated using the latest pattern. If a product was acquired **after** the creation date of the latest applicable XCA file, then there should not be any need for retro-calibration.

The IMAGE RETROCALIBRATION tool is applicable only to ASAR detected ground range products, which have the antenna pattern already applied and annotated and have been previously converted into power units.

The output image may have either a linear or dB scale.

It is also possible to apply a user generated antenna pattern.

AOI specification is permitted.

Example ".INI" file

```
[IMAGE RETROCALIBRATION]
Input Dir = "G:\backscattering\power\"
Output Dir = "G:\backscattering\out\"
Input Image = "power_IMP4399.APf"
Output Image Scale = "LINEAR"
Output Image = "power_IMP4399_out"
```

Parameter Summary: Image Retro-calibration

Input Image

The name of the input power image in internal format.
Example: `Input Image = "power_IMP4399.APf"`
mandatory INPUT
BEST extensions: ".APf", ".XTf", ".IT?", ".GT?", ".OV?", ".UNf", ".DBf", ".OP?", ".SGf", ".SGc", ".FI?", ".CR?", where "?" indicates that it is not important what format the data is in.

Output Image Scale

The scale of the output backscatter image:

- "LINEAR"
- "DB"

Do not use the dB scale if a further step of averaging is foreseen.
Example: `Output Image Scale = "LINEAR"`
mandatory INPUT

Output Image

The name of the output image in internal format containing the retro-calibrated data (the extension "*.BSf*" is automatically added by the system).
Example: `Output Image = "retro_IMP4399"`
mandatory INPUT
BEST extension: ".BSf"

Rough Range Calibration (ASAR)

Description

The ROUGH RANGE CALIBRATION tool corrects an image for the effect of incidence angle variation from near to far range, which is clearly visible in Wide Swath and Global Monitoring Mode products.

It can be applied either to amplitude or power images.

No AOI is permitted for this function.

Example ".INI" file

```
[ROUGH RANGE-CALIBRATION]
Input Dir = "C:\BEST_out\"
Output Dir = "C:\BEST_out\"
Input Image = "WSM.XTs"
Output Image = "WSM_rough"
```

Parameter Summary: Rough Range-Calibration

Input Image

The name of an ASAR WS or GM image in internal format. The input may be an amplitude or power image.
Example: `Input Image = "WSM.XTs"`
mandatory INPUT
BEST extension: ".APf", ".PAf", ".XTf", ".IT?", ".GT?", ".OV?", ".UNf", ".DBf", ".OP?", ".SGf", ".SGc", ".FI?", ".CR?", where "?" indicates that it is not important what format the data is in.

Output Image

The name of the output image in internal format containing the rough range calibrated data (the extension "*.XTf*" is automatically added by the system).
Example: `Output Image = "WSM_rough"`
mandatory OUTPUT
BEST extension: ".XTf"

Swath Enhancement (ASAR)

Description

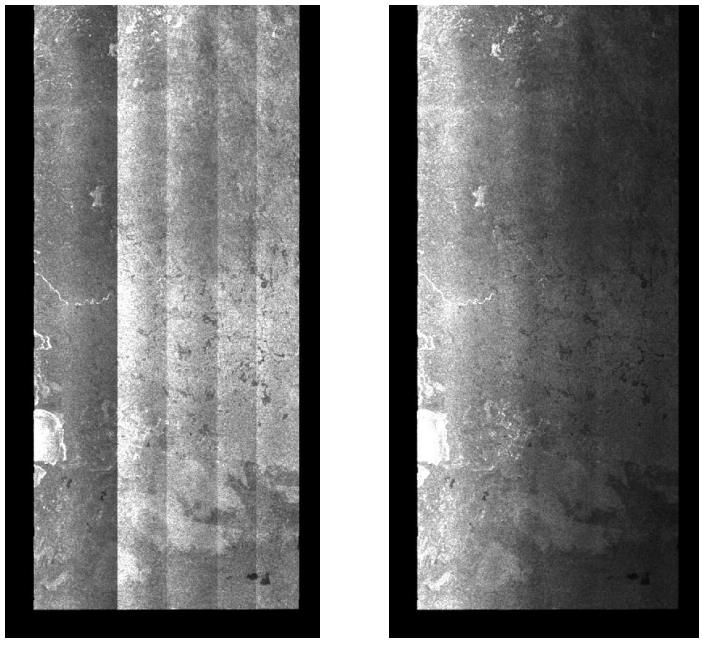
The SWATH ENHANCEMENT tool enables the user to correct ASAR Wide Swath and Global Monitoring Mode products affected by intensity discontinuities between sub-swaths. The resulting image will not be radiometrically sound, but gives a more aesthetically pleasing appearance.

The tool applies a linear coefficient, named Gain, to each of the five sub-swaths of the image. Both the Gains and the range limits of each sub-swath (in terms of their end column) must be provided by the user.

By virtue of the ScanSAR acquisition process, Wide Swath and Global Monitoring Mode products are made up of five independent swaths of imagery that may exhibit differing radiometry. Here, a methodology is suggested for manually evaluating a WS image to characterise the inter-swath differences and determine the gain values required by the tool:

- First, select (arbitrarily) a ‘master’ sub-swath, for which the gain value is set to one.
- Than, using image-processing software, find the relative intensity (a ratio) of the other sub-swaths by averaging over a homogenous area of the image and compute the inverse to enter as a Gain value.

The figures below show a Global Monitoring product before and after swath enhancement. The steps between adjacent sub-swaths visible on the left are reduced in the corrected image.



Example “INI” file

The following “*.ini*” file is an example of swath enhancement for a Global Monitoring product:

```
[ENHANCEMENT SWATH]
Input Dir = "C:\BEST_out\"
```

```
Output Dir = "C:\BEST_out\"
Input Image = "GM1.XTs"
Output Image = "GM1_enh"
SW1 Gain = 5.4925
SW2 Gain = 2.197
SW3 Gain = 1.69
SW4 Gain = 1.3
SW5 Gain = 1
SW1 end col = 348
SW2 end col = 500
SW3 end col = 658
SW4 end col = 778
```

Parameter Summary: Swath Enhanacement

Input Image

The name of an ASAR WS or GM amplitude image in internal format.
Example: `Input Image = "GM1.XTs"`
mandatory INPUT
BEST extension:

Output Image

The name of the output amplitude image in internal format containing the enhanced data (the extension "*.XT?*" is automatically added by the system).
Example: `Output Image = "GM1_enh"`
mandatory OUTPUT
BEST extension: ".XT?"

SWn Gain (n = 1, 2, 3, 4, 5)

The linear gain to be applied to swath 'n'.
Example: `SW1 Gain = 5.4925`
mandatory parameter

SWn end col (n = 1, 2, 3, 4)

The number (counting from near range) of the last column in swath 'n'.
Example: `SW1 end col = 348`
mandatory parameter

C APPENDICES

Appendix 1: Example of a Header Analysis output file

An example of the ASCII Header Analysis output file is shown here. However, these files are very long so only a subsection is shown here to provide an example of its format.

The following information is given in the six columns:

- parameter sequential number
- name of the field as it appears in the ESA format documentation
- value of the parameter
- units in which the parameter is expressed
- internal field name, as it appears in the parameter dump obtained with the data conversion tool
- a remark

```
================================================================================================================================================================
BEST - ESA / Telespazio - ANNOTATION LIST
================================================================================================================================================================
Processing time.............: 29-Mar-2005 12:11:26.000
Product type................: slc
Sensor Mode.................: Image
Source......................: ASAR
Data format.................: MPH-SPH
Facility id.................: esp
Format descriptor record...: C:\Software\BEST\\cfg\slc3eespim
----------------------------------------------------------------------------------------------------------------------------------------------------------------
File name...................: PDF - PRODUCT_DATA_FILE
Record name.................: Main Product Header Record
 Pos Esa field name                          Value                       Units           Tag                       Remark
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   1 dummy                                   PRODUCT="                   -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   2 Product Tag                             ASA_IMS_1PNUPA20050122_09   -               product_name              contains the string 'PRODUCT="
                                             5556_000000162034_00065_1                                             '
                                             5149_0972.N1
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   3 Product ID                              ASA_IMS_1P                  -               envisat_prod_id           -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   4 Processing State Flag                   N                           -               envisat_proc_state_flag   should be equal to the one bel
                                                                                                                   ow
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   5 Originator ID                           UPA                         -               envisat_originator        -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   6 Start Day                               20050122                    -               envisat_start_day         Start day of first MDSR, or fi
                                                                                                                   le creation date for aux files
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   7 dummy                                   _                           -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   8 Start Time                              095556                      -               envisat_start_time        Start time of first MDSR, or f
                                                                                                                   ile creation time for aux file
                                                                                                                   s
----------------------------------------------------------------------------------------------------------------------------------------------------------------
   9 dummy                                   _                           -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  10 Duration                                00000016                    seconds         envisat_duration          -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  11 Phase ID                                2                           -               envisat_phase_id          'X' if not used
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  12 Cycle Number within the phase           034                         -               envisat_cycle_no          -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  13 dummy                                   _                           -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  14 Orbit Number relative to the start of pr 00065                      -               orbit_num                 -
     oduct
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  15 dummy                                   _                           -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  16 Absolute Orbit Number                   15149                       -               envisat_absolute_orbit_no -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  17 dummy                                   _                           -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  18 Product Type File Counter               0972                        -               envisat_prod_ty_file_coun 0000 to 9999, then wraps to 00
                                                                                         ter                       00
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  19 Period                                  .                           -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  20 Satellite ID                            N1                          -               envisat_sat_id            ENVISAT-1=N1, ERS1=E1, ERS2=E2
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  21 dummy                                   "                           -               -                         '"' and '\n'
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  22 Processing Stage Flag                   PROC_STAGE=N                -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  23 Reference Document Describing Product   REF_DOC="PO-RS-MDA-GS2009    -               -                         -
                                             _08_3H "
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  24 Spare                                   -                           -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  25 Acquisition Station                     ACQUISITION_STATION="PDAS   -               -                         -
                                             -F            "
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  26 Processing Center Tag                   PROC_CENTER="               -               -                         -
----------------------------------------------------------------------------------------------------------------------------------------------------------------
  27 Processing Center ID                    UK-PAC                      -               processing_paf            Processing Center which genera
                                                                                                                   ted this product
```

28 dummy	"	-	-	-
29 Processing UTC Time	PROC_TIME="15-FEB-2005 12 :01:04.000000"	UTC	-	if not used, it's all spaces
30 Software Version Tag	SOFTWARE_VER="	-	-	-
31 Software version of processing software	ASAR/3.08	-	processor_name	-
32 dummy	"	-	-	-
33 Spare	-	-	-	-
34 UTC Start Time of data sensing	SENSING_START="22-JAN-200 5 09:55:56.061990"	UTC	-	for Level0 products use this v alue, for Level1 use the one i n the SPH
35 UTC Stop Time of data sensing	SENSING_STOP="22-JAN-2005 09:56:13.061654"	UTC	-	for Level0 products use this v alue, for Level1 use the one i n the SPH
36 Spare	-	-	-	-
37 Phase letter	PHASE=2	-	-	-
38 Cycle	CYCLE=+034	-	-	-
39 Relative Orbit Number	REL_ORBIT=+00065	-	-	-
40 Absolute Orbit Number	ABS_ORBIT=+15149	-	-	-
41 State Vector Time TAG	STATE_VECTOR_TIME="	-	-	-
42 UTC of ENVISAT state vector	22-JAN-2005 09:55:00.0000 00	UTC	ascend_node_utc_time	-
43 dummy	"	-	-	-
44 Delta UT1	DELTA_UT1=-.517329<s>	seconds	-	Delta UT1 = UT1-UTC
45 X_POSITION TAG	X_POSITION=	-	-	-
46 X Position in Earth-Fixed Reference	+4841205.128	meters	ascend_node_x	-
47 unit specifier	<m>	-	-	<m>
48 dummy	-	-	-	-
49 Y_POSITION TAG	Y_POSITION=	-	-	-
50 Y Position in Earth-Fixed Reference	+0891365.204	meters	ascend_node_y	-
51 unit specifier	<m>	-	-	<m>
52 dummy	-	-	-	-
53 Z_POSITION TAG	Z_POSITION=	-	-	-
54 Z Position in Earth-Fixed Reference	+5196193.458	meters	ascend_node_z	-
55 unit specifier	<m>	-	-	<m>
56 dummy	-	-	-	-
57 X_VELOCITY TAG	X_VELOCITY=	-	-	-
58 X Velocity in Earth-Fixed Reference	+5565.602399	m/sec	ascend_node_vx	-
59 unit specifier	<m/s>	-	-	<m/s>
60 dummy	-	-	-	-
61 Y_VELOCITY TAG	Y_VELOCITY=	-	-	-
62 Y Velocity in Earth-Fixed Reference	-0981.572338	m/sec	ascend_node_vy	-
63 unit specifier	<m/s>	-	-	<m/s>
64 dummy	-	-	-	-
65 Z_VELOCITY TAG	Z_VELOCITY=	-	-	-
66 Z Velocity in Earth-Fixed Reference	-5004.591697	m/sec	ascend_node_vz	-
67 unit specifier	<m/s>	-	-	-
68 dummy	-	-	-	-
69 Source of Orbit Vectors	VECTOR_SOURCE="FR"	-	-	-
70 Spare	-	-	-	-
71 UTC time corresponding to SBT below	UTC_SBT_TIME="22-JAN-2005 09:41:31.323738"	UTC	-	-
72 SAT_BINARY_TIME TAG	SAT_BINARY_TIME=	-	-	-
73 Satellite Binary Time	+1939849984	-	satellite_bin_time_code	32bit integer time of satellit e clock. Zero if not used
74 dummy	-	-	-	-
75 Clock Step Size	CLOCK_STEP=+3906249800<ps >	psec	-	expressed in picoseconds. If n ot used is set to all zeroes
76 Spare	-	-	-	-
77 UTC time of the occurrence of the Leap S econd	LEAP_UTC="17-OCT-2001 00: 00:00.000000"	UTC	-	All spaces if not used
78 Leap Second Sign	LEAP_SIGN=+001	-	-	+001 if positive, -001 if nega tive, +000 if not used
79 Leap second error	LEAP_ERR=0	-	-	if leap second occurs =1, othe rwise 0; if not used =0
80 Spare	-	-	-	-
81 Product Error	PRODUCT_ERR=1	-	-	1 if there are errors - user s hould check SPH or Summary Qua lity ADS for details.

82 Total Size of Product	TOT_SIZE=+000000000005571 38358<bytes>	bytes	-	-
83 Length of SPH	SPH_SIZE=+0000006099<byte s>	bytes	-	-
84 Number of DSDs	NUM_DSD=+0000000018	-	-	-
85 Length of each DSD	DSD_SIZE=+0000000280<byte s>	bytes	-	-
86 Number of DSDs attached	NUM_DATA_SETS=+0000000006	-	-	-
87 Spare	-	-	-	-

Record name.....: Specific Product Header Head Record

Pos Esa field name	Value	Units	Tag	Remark
1 SPH Descriptor	SPH_DESCRIPTOR="Image Mod e SLC Image "	-	-	-
2 Stripline Continuity Indicator	STRIPLINE_CONTINUITY_INDI CATOR=+000	-	-	0 if the product is a comlete segment; otherwise: stripline counter
3 Slice position	SLICE_POSITION=+001	-	-	from +001 to stripline continu ity, default is +001
4 Number of slices in this stripline	NUM_SLICES=+	-	-	default if no continuity: +001
5 Number of slices in this stripline	001	-	num_slices	default if no continuity: +001
6 dummy	-	-	-	-
7 FIRST_LINE_TIME TAG	FIRST_LINE_TIME="	-	-	-
8 First Zero Doppler Azimuth time of produ ct	22-JAN-2005 09:55:56.5170 81	UTC	zero_dopp_azim_first_time	UTC of 1st range line in the M DS of this product
9 dummy	"	-	-	-
10 LAST_LINE_TIME TAG	LAST_LINE_TIME="	-	-	-
11 Last Zero Doppler Azimuth time of produc t	22-JAN-2005 09:56:12.7908 31	UTC	zero_dopp_azim_last_time	UTC of last range line in the MDS of this product
12 dummy	"	-	-	-
13 FIRST_NEAR_LAT TAG	FIRST_NEAR_LAT=	-	-	-
14 Geodetic latitude of the first sample at the first line	+0043932217	10^-6 deg	top_left_lat	-
15 unit specifier	<10-6degN>	-	-	<10-6degN>
16 dummy	-	-	-	-
17 FIRST_NEAR_LONG TAG	FIRST_NEAR_LONG=	-	-	-
18 East geodetic longitude of the first sam ple of the first line	+0006346123	10^-6 deg	top_left_lon	-
19 unit specifier	<10-6degE>	-	-	<10-6degE>
20 dummy	-	-	-	-
21 Geodetic Latitude of the middle sample o f the 1st line	FIRST_MID_LAT=+0044042161 <10-6degN>	10^-6 deg	-	-
22 East geodetic longitude of the middle sa mple of the first line	FIRST_MID_LONG=+000565383 5<10-6degE>	10^-6 deg	-	-
23 FIRST_FAR_LAT TAG	FIRST_FAR_LAT=	-	-	-
24 Geodetic Latitude of the last sample of the first line	+0044132757	10^-6 deg	top_right_lat	-
25 unit specifier	<10-6degN>	-	-	<10-6degE>
26 dummy	-	-	-	-
27 FIRST_FAR_LONG TAG	FIRST_FAR_LONG=	-	-	-
28 East geodetic longitude of the last samp le of the first line	+0005060429	10^-6 deg	top_right_lon	-
29 unit specifier	<10-6degE>	-	-	<10-6degE>
30 dummy	-	-	-	-
31 LAST_NEAR_LAT TAG	LAST_NEAR_LAT=	-	-	-
32 Geodetic Latitude of the first sample of the last line	+0042977177	10^-6 deg	bottom_left_lat	-
33 unit specifier	<10-6degN>	-	-	<10-6degN>
34 dummy	-	-	-	-
35 LAST_NEAR_LONG TAG	LAST_NEAR_LONG=	-	-	-
36 East geodetic longitude of the first sam ple of the last line	+0006045342	10^-6 deg	bottom_left_lon	-
37 unit specifier	<10-6degE>	-	-	<10-6degE>
38 dummy	-	-	-	-
39 Geodetic Latitude of the middle sample o f the last line	LAST_MID_LAT=+0043086310< 10-6degN>	10^-6 deg	-	-
40 East geodetic longitude of the middle sa mple of the last line	LAST_MID_LONG=+0005365082 <10-6degE>	10^-6 deg	-	-
41 LAST_FAR_LAT TAG	LAST_FAR_LAT=	-	-	-
42 Geodetic Latitude of the last sample of the last line	+0043176387	10^-6 deg	bottom_right_lat	-
43 unit specifier	<10-6degN>	-	-	<10-6degN>
44 dummy	-	-	-	-

45	LAST_FAR_LONG TAG	LAST_FAR_LONG=	-	-	-
46	East geodetic longitude of the last sample of the last line	+0004781699	10^-6 deg	bottom_right_lon	-
47	unit specifier	<10-6degE>	-	-	<10-6degE>
48	dummy	-	-	-	-
49	Spare	-	-	-	-
50	Swath number	SWATH="	-	-	-
51	Swath number	IS2	-	swath_number	-
52	Swath number	"	-	-	-
53	Ascending or descending orbit designator	PASS="DESCENDING"	-	-	"ASCENDING ","DESCENDING" or " FULL ORBIT"
54	SAMPLE_TYPE TAG	SAMPLE_TYPE="	-	-	-
55	Detected or complex sample type designator	COMPLEX	-	envisat_sampletype	"DETECTED" or "COMPLEX "
56	dummy	"	-	-	-
57	Processing Algorithm used	ALGORITHM="RAN/DOP"	-	-	"RAN/DOP" or "SPECAN "
58	Processing Algorithm used	MDS1_TX_RX_POLAR="	-	-	"RAN/DOP" or "SPECAN "
59	Transmitter/Receiver Polarization for MDS 1	V/V	-	polarization_1	-
60	Processing Algorithm used	"	-	-	"RAN/DOP" or "SPECAN "
61	Processing Algorithm used	MDS2_TX_RX_POLAR="	-	-	"RAN/DOP" or "SPECAN "
62	Transmitter/Receiver Polarization for MDS 2	-	-	polarization_2	-
63	Processing Algorithm used	"	-	-	"RAN/DOP" or "SPECAN "
64	Compression algorithm used on echo data on-board the satellite	COMPRESSION="FBAQ4"	-	-	-
65	AZIMUTH_LOOKS TAG	AZIMUTH_LOOKS=	-	-	-
66	Number of Looks in Azimuth	+001	-	nom_nb_looks_azim	-
67	dummy	-	-	-	-
68	RANGE_LOOKS TAG	RANGE_LOOKS=	-	-	-
69	Number of Looks in Range	+001	-	nom_nb_looks_range	-
70	dummy	-	-	-	-
71	RANGE_SPACING TAG	RANGE_SPACING=	-	-	-
72	Range sample spacing in meters	+7.80397463E+00	meters	pixel_spacing	-
73	unit specifier	<m>	-	-	<m>
74	dummy	-	-	-	-
75	AZIMUT_SPACING TAG	AZIMUTH_SPACING=	-	-	-
76	Nominal azimuth sample spacing in meters	+4.04308319E+00	meters	line_spacing	-
77	unit specifier	<m>	-	-	<m>
78	dummy	-	-	-	-
79	Azimuth sample spacing in time (Line Time Interval)	LINE_TIME_INTERVAL=+6.051 74631E-04<s>	seconds	-	-
80	LINE_LENGTH TAG	LINE_LENGTH=	-	-	-
81	Number of samples per output line	+05175	-	image_width	includes zero filled samples; for complex images, 1 sample is a I,Q pair
82	unit specifier	<samples>	-	-	<samples>
83	dummy	-	-	-	-
84	DATA_TYPE TAG	DATA_TYPE="	-	-	-
85	Output data type	SWORD	-	envisat_datatype	-
86	dummy	"	-	-	-
87	Spare	-	-	-	-

Appendix 2: Example of a Media Analysis output file

An example of the ASCII Media Content Report (MCR file) is shown here.

The media analysis file is divided into two sections. In the first, a list of recognised products is given with the three best choices (the most likely first). In the second section, the media structure is shown, detailing the number of volumes, files and records and their size in bytes.

```
-------------------------------------------------------------
Number of Volume(s) = 1
Product Type        = PRI
Sensor Id           = ERS2
Data Format         = CEOS
Source Id           = DEP
-------------------------------------------------------------
Number of Volume(s) = 1
Product Type        = PRI
Sensor Id           = ERS1
Data Format         = CEOS
Source Id           = ESP
-------------------------------------------------------------
Number of Volume(s) = 1
Product Type        = PRI
Sensor Id           = ERS2
Data Format         = CEOS
Source Id           = ESP
-------------------------------------------------------------
Number of Volume(s) = 1
-------------------------------------------------------------
VOLUME: 1

FILE: 1   Number of Record(s)    Record Size
RECORD:                     4            360

FILE: 2   Number of Record(s)    Record Size
RECORD:                     1            720
RECORD:                     1           1886
RECORD:                     1           1620
RECORD:                     1           1046
RECORD:                     2          12288

FILE: 3   Number of Record(s)    Record Size
RECORD:                  8202          16012

FILE: 4   Number of Record(s)    Record Size
RECORD:                     1            360
-------------------------------------------------------------
```

Appendix 3: Ancillary Data Dump Output and Annotations

An example of an ancillary data dump is shown here.

```
[ANNOTATIONS]
 Image = C:\Data\ASAR\ASA_IMS.XTt
 image_width =          5175
 image_length =         26892
 bits_per_sample =            VectorialTag[0]=16
 compression =          1
 photometric_interpretation =            1
 sample_per_pixel =           2
 x_print_resolution =         300.000000
 y_print_resolution =         300.000000
 resolution_unit =            2
 tile_width =           128
 tile_length =          128
 tile_offset =          VectorialTag[0]=69232
 tile_byte_count =            VectorialTag[0]=65536
 sample_format =        VectorialTag[0]=2
 disposition =          x
 absolute_calib_k =           34994.515625
 antenna_boresight =          0.000000
 antenna_elevation_gain_flag =           0
 bottom_left_lat =            42.977173
 bottom_left_lon =            6.045401
 bottom_right_lat =           43.176392
 bottom_right_lon =           4.781658
 centre_geodetic_lat =        43.563950
 centre_geodetic_lon =        5.510607
 early_zero_fill_record_number =         0
 late_zero_fill_record_number =          0
 cross_dopp_freq_const =            +294.598663
 cross_dopp_freq_linear =           -966150.437500
 cross_dopp_freq_quad =             +1370826624.000000
 day_data_point =             22
 ellipsoid_semimajor_axis =         6378.137207
 ellipsoid_semiminor_axis =         6356.752441
 incid_angle_centre_range =         0.000000
 line_spacing =         4.043083
 map_proj_descr =             Slant range
 month_data_point =           1
 nom_nb_looks_azim =          1.000000
 nom_nb_looks_range =         1.000000
 normalisation_ref_range =          800.000000
 orbit_num =            15149
 pixel_spacing =        7.803975
 processor_range_compression =            NOMINAL
 radar_wavelen =        0.056236
 replica_power =        2.833618
 sampling_rate =        19.207680
 scene_ref_num =        ORBIT=15149
 second_of_day =        35756.515625
 sensor_plat_mission_id =
 time_interval_data_point =         +4.068438
```

```
top_left_lat =         43.932220
top_left_lon =         6.346062
top_right_lat =        44.132755
top_right_lon =        5.060472
year_data_point =            2005
zero_dopp_azim_first_time =        22-JAN-2005 09:55:56.517
zero_dopp_azim_last_time =         22-JAN-2005 09:56:12.790
zero_dopp_range_first_time =             5.529571
int_top_left_east =          0.000000
int_top_left_north =         0.000000
int_top_right_north =        0.000000
int_top_right_east =         0.000000
int_bottom_left_east =             0.000000
int_bottom_left_north =            0.000000
int_bottom_right_north =           0.000000
int_bottom_right_east =            0.000000
input_columns_nb =           0
input_lines_nb =             0
spread_loss_comp_flag =            0
nb_data_points =             5
log_vol_id =           ENVI.ASA.SLC
gr_sr_pol_degree =           0
near_zero_fill_pixel_number =            0
far_zero_fill_pixel_number =             0
orbit_direction =            DESCENDING
prf =           1652.415649
cross_dopp_freq_quartic =          +0.000000
envisat_first_vect_mjd_days =            1848
envisat_2nd_vect_mjd_days =        1848
envisat_first_vect_mjd_seconds =         35756
envisat_first_vect_mjd_microsec =        517081
envisat_2nd_vect_mjd_seconds =           35760
envisat_2nd_vect_mjd_microsec =          585519
envisat_3rd_vect_mjd_days =        1848
envisat_3rd_vect_mjd_seconds =           35764
envisat_3rd_vect_mjd_microsec =          653956
envisat_4th_vect_mjd_days =        1848
envisat_4th_vect_mjd_seconds =           35768
envisat_4th_vect_mjd_microsec =          722394
envisat_5th_vect_mjd_days =        1848
envisat_5th_vect_mjd_seconds =           35772
envisat_5th_vect_mjd_microsec =          790831
envisat_source_file =ASA_IMS_1PNUPA20050122_095556_000000162034_00065...
ls_en_conv_coeff_1 =         0.000000e+000
ls_en_conv_coeff_2 =         0.000000e+000
ls_en_conv_coeff_3 =         0.000000e+000
ls_en_conv_coeff_4 =         0.000000e+000
ls_en_conv_coeff_5 =         0.000000e+000
ls_en_conv_coeff_6 =         0.000000e+000
ls_en_conv_coeff_7 =         0.000000e+000
ls_en_conv_coeff_8 =         0.000000e+000
en_ls_conv_coeff_1 =         0.000000e+000
en_ls_conv_coeff_2 =         0.000000e+000
en_ls_conv_coeff_3 =         0.000000e+000
en_ls_conv_coeff_4 =         0.000000e+000
en_ls_conv_coeff_5 =         0.000000e+000
en_ls_conv_coeff_6 =         0.000000e+000
en_ls_conv_coeff_7 =         0.000000e+000
en_ls_conv_coeff_8 =         0.000000e+000
actual_product_type =        SLC
geolocationgrid_tiepoints =        11
```

```
geolocationgrid_1stlinenum =             1, 2446, 4891, 7336, 9781...
geolocationgrid_totlinenum =             2445, 2445, 2445, 2445, 2445...
geolocationgrid_samplenum =        1, 519, 1037, 1555, 2073, 2588, 3109...
geolocationgrid_slanttime =        +5528476.500000, +5555445.000000...
geolocationgrid_incangle =         +18.670341, +19.583149, +20.449125...
geolocationgrid_latitude =         43932217, 43956374, 43979317, 44001203...
geolocationgrid_longitude =        6346123, 6196467, 6053077, 5915136...
azimuth_time_grid_mjd_days =             1848, 1848, 1848, 1848, 1848...
azimuth_time_grid_mjd_seconds =          35756, 35757, 35759, 35760...
azimuth_time_grid_mjd_microsec =         517081, 996733, 476385, 956037...
attachment_flag_grid =             0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
subsatellite_track_heading =             -165.435593, -165.448471...
product_name =         ASA_IMS_1PNUPA20050122_095556_000000162034_00065...
product_error =
swath_number =         IS2
polarization_1 =             V/V
platform_h =
num_slices =           1
azimuth_time_dopp_mjd_days =             1848
azimuth_time_dopp_mjd_seconds =          35764
azimuth_time_dopp_mjd_microsec =         561969
absolute_calib_k2 =          0.000000
x_sat_1 =        514699593
x_sat_2 =        516831736
x_sat_3 =        518954381
x_sat_4 =        521067490
x_sat_5 =        523171021
y_sat_1 =        83311311
y_sat_2 =        82871030
y_sat_3 =        82428006
y_sat_4 =        81982252
y_sat_5 =        81533782
z_sat_1 =        490448948
z_sat_2 =        488282118
z_sat_3 =        486106498
z_sat_4 =        483922126
z_sat_5 =        481729041
vx_sat_1 =             525233184
vx_sat_2 =             522903647
vx_sat_3 =             520564337
vx_sat_4 =             518215310
vx_sat_5 =             515856597
vy_sat_1 =             -107880440
vy_sat_2 =             -108556364
vy_sat_3 =             -109228969
vy_sat_4 =             -109898234
vy_sat_5 =             -110564138
vz_sat_1 =             -531511438
vz_sat_2 =             -533676966
vz_sat_3 =             -535832902
vz_sat_4 =             -537979220
vz_sat_5 =             -540115871
subimg_top_left_row =        0
subimg_top_left_col =        0
proc_history =   HEADER DECODE 29-Mar-2005 12:11:27.000, FULL RESOLUTION 29-
     Mar-2005 16:10:46.000
pixel_type =           COMPLEX
calib_const_appli_flag =           0
adc_satur_compens_flag =           0
chirp_average_density =            0.000000
processing_paf =             unknown
```

```
processor_name =             ASAR
scaling_factor =             1.000000
x_scale_factor =             1.000000
y_scale_factor =             1.000000
prf_equivalent =             1652.491821
prf_equivalent_full =        1652.491821
doppl_centr_cub_coeff =            +0.000000
replica_power_comp_flag =          0
image_scale =          LINEAR
data_format =          mph-sph
source_id =            esp
number_of_volumes =          1
row_transient =        0
col_transient =        0
presentation =         NORMAL
nominal_replica_comp_flag =        0
dopp_freq_degree =           4
sensor_mode =          image
full_image_length =          26892
full_image_width =           5175
```

The following table explains the annotations maintained by the BEST tools:

Annotation name	Meaning	Example value
absolute_calib_k	calibration constant value (linear)	999978.000000
adc_satur_compens_flag	flag indicating if the ADC saturation compensation has been applied (1 means applied)	0
antenna_boresight	boresignt angle (degrees)	20.355000
antenna_elevation_gain_flag	flag indicating if the antenna pattern correction is apllied (1 means applied)	0
bits_per_sample	the number of bits of the pixel of each layer of the image	32
bottom_left_lat	latitude of the last line first pixel corner (degree)	51.914104
bottom_left_lon	longitude of the last line first pixel corner (degree)	6.062989
bottom_right_lat	latitude of the last line last pixel corner (degree)	51.925541
bottom_right_lon	longitude of the last line last pixel corner (degree)	5.985325
calib_const_appli_flag	flag indicating if the spreading calibtration constant has been applied (1 means applied)	0
centre_geodetic_lat	latitude of the center (degree)	52.349888
centre_geodetic_lon	latitude of the center (degree)	6.195046
chirp_average_density	density of the chirp replica	0.000000
col_transient	internal TTIF flag	0
compression	internal TTIF flag	1
cross_dopp_freq_const	doppler frequency polynomial order 0 coefficient (Hz)	150.656296

cross_dopp_freq_linear	doppler frequency polynomial order 1 coefficient (Hz/ s)	61187.777344
cross_dopp_freq_quad	doppler frequency polynomial order 2 coefficient (Hz/s/ s)	0.000000
data_format	format of the SAR product (CEOS or MPHSPH)	CEOS
day_data_point	day in the year of the first state vector	4
disposition	internal TTIF flag	x
early_zero_fill_record_numb er	number of fill lines at image start	0
ellipsoid_semimajor_axis	ellipsoid semimajor axis (km)	6378.144043
ellipsoid_semiminor_axis	ellipsoid semiminor axis (km)	6356.758789
far_zero_fill_pixel_number	number of filled pixels at end of each image line	68
gr_sr_coeff_1	slant to ground polynomial coeffient 1	0.000969
gr_sr_coeff_2	slant to ground polynomial coeffient 2	526.446899
gr_sr_coeff_3	slant to ground polynomial coeffient 3	11.997012
gr_sr_coeff_4	slant to ground polynomial coeffient 4	-0.061625
gr_sr_coeff_5	slant to ground polynomial coeffient 5	-0.000199
gr_sr_pol_degree	slant to ground polynomial degree	4
image_length	the number of lines of the image	25
image_scale	indication if the image is in LINEAR or DB scale	LINEAR
image_width	the number of pixels of the image	45
incid_angle_centre_range	incidence angle at mid range (degrre)	23.069057
late_zero_fill_record_number	number of fill lines at image end	0
line_spacing	spacing between lines (m)	150.000000
log_vol_id	product identifier string	ERS2.SAR.PRI
map_proj_descr	descriptor of the geographic projection	Ground range
month_data_point	month in the year of the first state vector	8
nb_data_points	number of the state vectors	5
near_zero_fill_pixel_number	number of filled pixels at start of each image line	0
nom_nb_looks_azim	number of looks	3.000000
normalisation_ref_range	reference slant range used for the spreading loss	847.000000

	compensation (km)	
number_of_volumes	number of media volumes	1
photometric_interpretation	internal TTIF flag	1
pixel_spacing	spacing between pixels (m)	125.000000
pixel_type	identificator of the Amplitude or Power or Complex image	Amplitude
prf	Pulse Repetition Frequency (Hz)	1679.902344
prf_equivalent	internal TTIF flag	19.524895
proc_history	processing history sequence	
processing_paf	identification of the processing station/PAF	IP
processor_name	identification of the SAR processing system	SAR ERS
radar_wavelen	wavelenght of the radar signal (m)	0.056565
replica_power	power of the replica chirp	154641.000000
resolution_unit	internal TTIF flag	2
row_transient	internal TTIF flag	0
sample_format	format of the pixel: 1,2 means integer, 4 means floating point representation	4
sample_per_pixel	number of image layers	1
sampling_rate	sampling frequency in range (MHz)	18.959999
scaling_factor	internal TTIF flag	1.000000
scene_ref_num	scene identification string	ORBIT: 1508 - FRAME: 2547
second_of_day	second in the day of the first state vector (s)	37980.000000
source_id	generating station/PAF	IP
spread_loss_comp_flag	flag indicating if the spreading loss compensation has been applied (1 means applied)	1
subimg_top_left_col	first line first pixel corner in the entire image coordinate system (column value)	2
subimg_top_left_row	first line first pixel corner in the entire image coordinate system (line value)	3
tile_byte_count	internal TTIF flag	65536
tile_length	internal TTIF flag	128
tile_offset	internal TTIF flag	24
tile_width	internal TTIF flag	128
time_interval_data_point	number of seconds between contiguous state vectors (s)	60.000000
top_left_lat	latitude of the first line first pixel corner (degree)	52.773964
top_left_lon	longitude of the first line first	6.408101

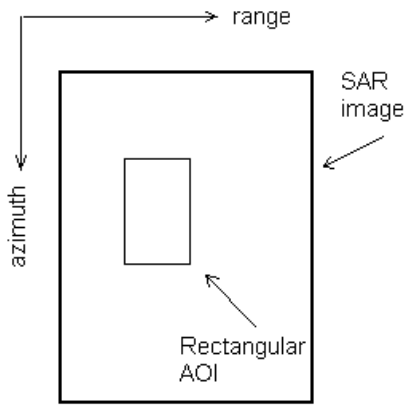
	pixel corner (degree)	
top_right_lat	latitude of the first line last pixel corner (degree)	52.785542
top_right_lon	longitude of the first line last pixel corner (degree)	6.328794
vx_sat_1	state vector velocity 1 (x component)	6535.191690
vx_sat_2	state vector velocity 2 (x component)	6283.327860
vx_sat_3	state vector velocity 3 (x component)	6006.074500
vx_sat_4	state vector velocity 4 (x component)	5704.555740
vx_sat_5	state vector velocity 5 (x component)	5379.997550
vy_sat_1	state vector velocity 1 (y component)	-887.918640
vy_sat_2	state vector velocity 2 (y component)	-999.248640
vy_sat_3	state vector velocity 3 (y component)	-1104.391740
vy_sat_4	state vector velocity 4 (y component)	-1202.721190
vy_sat_5	state vector velocity 5 (y component)	-1293.646730
vz_sat_1	state vector velocity 1 (z component)	-3664.545220
vz_sat_2	state vector velocity 2 (z component)	-4057.490410
vz_sat_3	state vector velocity 3 (z component)	-4434.628530
vz_sat_4	state vector velocity 4 (z component)	-4794.478790
vz_sat_5	state vector velocity 5 (z component)	-5135.625370
x_print_resolution	internal TTIF flag	300.000000
x_sat_1	state vector 1 (x component)	3569723.600000
x_sat_2	state vector 2 (x component)	3954408.780000
x_sat_3	state vector 3 (x component)	4323215.080000
x_sat_4	state vector 4 (x component)	4674652.340000
x_sat_5	state vector 5 (x component)	5007300.920000
x_scale_factor	internal TTIF flag	0.100000
y_print_resolution	internal TTIF flag	300.000000
y_sat_1	state vector 1 (y component)	880854.570000
y_sat_2	state vector 2 (y component)	824210.230000
y_sat_3	state vector 3 (y component)	761068.490000
y_sat_4	state vector 4 (y component)	691819.530000
y_sat_5	state vector 5 (y component)	616890.060000

y_scale_factor	internal TTIF flag	0.083333
year_data_point	year of the first state vector	1995
z_sat_1	state vector 1 (z component)	6138981.080000
z_sat_2	state vector 2 (z component)	5907244.760000
z_sat_3	state vector 3 (z component)	5652398.400000
z_sat_4	state vector 4 (z component)	5375435.140000
z_sat_5	state vector 5 (z component)	5077435.060000
zero_dopp_azim_first_time	time of the first image line	04-AUG-1995 10:35:02.322
zero_dopp_azim_last_time	time of the last image line	04-AUG-1995 10:35:17.687
zero_dopp_range_first_time	time of the first image pixel (ms)	5.564405

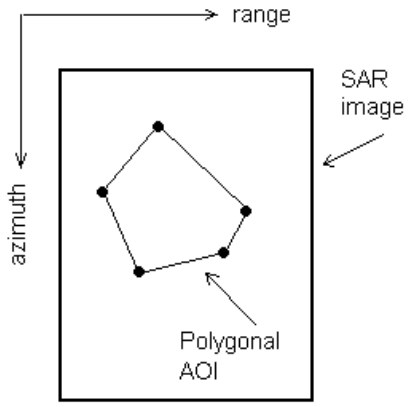
Appendix 4: AOI Specification

An AOI specified using coordinates, can have the following two shapes:

a rectangular region
a polygonal region



A Rectangular AOI

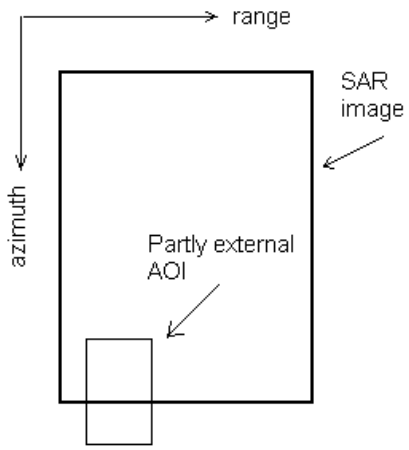


A Polygonal AOI

The AOI can be placed in two ways respect the SAR image:

internal
partly external

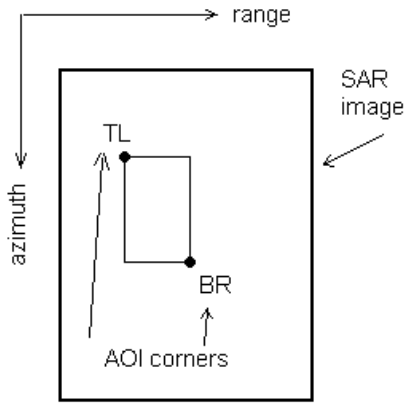
The figure “a Rectangular AOI” is related to a internal AOI while the following figure “AOI partly external to the image” shows an AOI placed partly externally to the image.



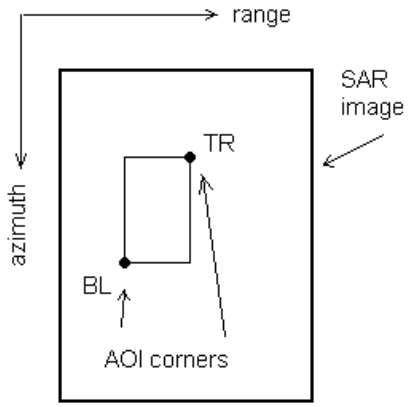
An AOI partly external to the image

The Rectangular AOI can be specified in the following forms:

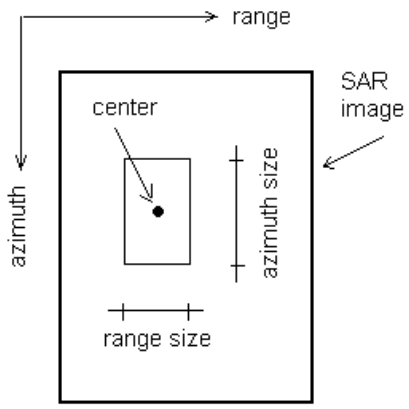
Top Left Corner and Bottom Right Corner
Top Right Corner and Bottom Left Corner
Center and Size



Rectangular AOI specified through Top Left, Bottom Right Corners



Rectangular AOI specified through Top Right, Bottom Left Corners



Rectangular AOI specified through Center, Size

Both the Top Left, Bottom Right Corners couple and the Top Right, Bottom Left Corners can be specified in one of the two coordinate systems:

geodetic latitude, longitude
row,column

The Center point can be specified in one of the two coordinate systems:

geodetic latitude, longitude
row,column

The Size can be specified in

kilometers unit
pixel units

The polygonal AOI can be specified through the following data:

the number of the vertices of the polygon
the coordinates of the vertices

The vertices can be expressed in the following coordinate systems:

geodetic latitude, longitude
row,column

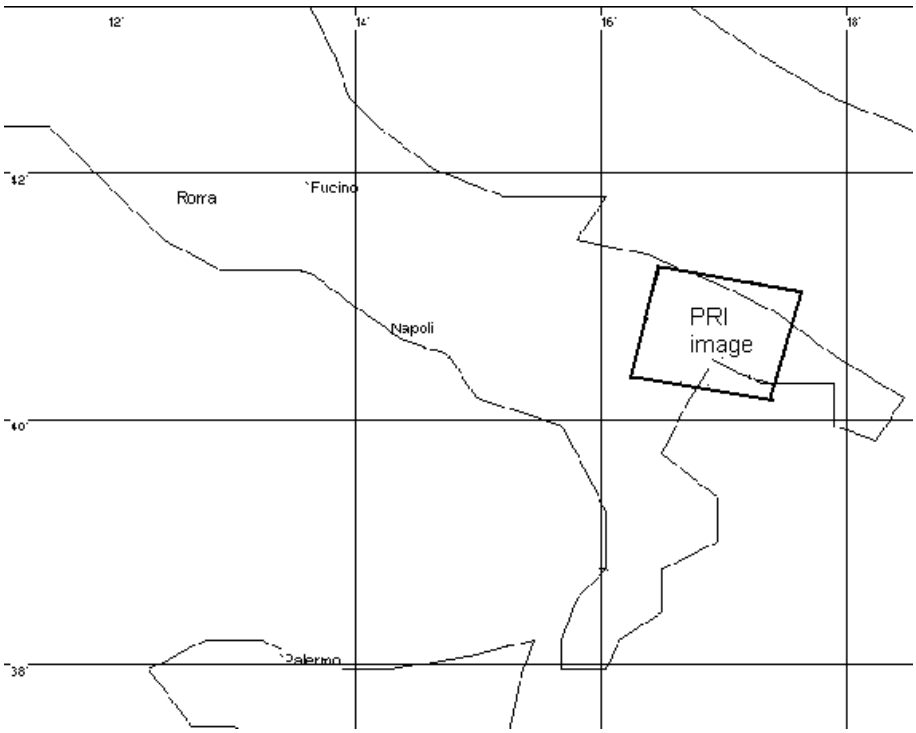
Note that the polygon area is built following the specified order of the vertices.

The following table summarizes the various specifications of the AOI with the related portions of the ".INI" file.

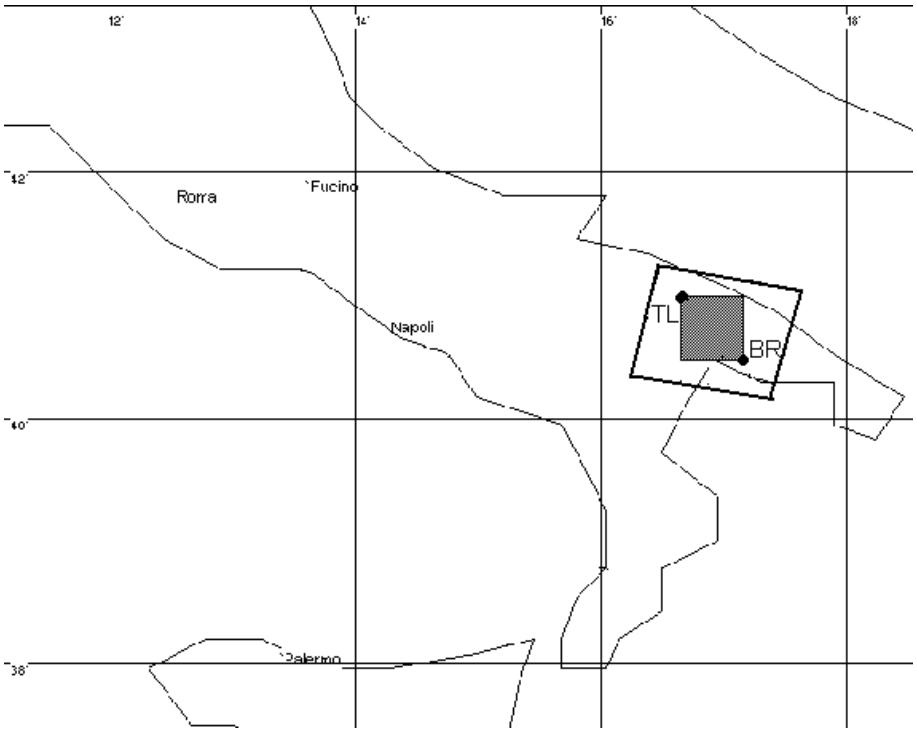
AOI Specification	Example	Comment
Rectangular Top Left Bottom Right cor ners Row, Column	Coordinate System="ROWCOL" Top Left Corner=100,200 Bottom Right Corner=300,500	the first coordinate is for row, the second for column the row,col coordinate system is assumed by default if the

coordinates		Coor dinate System statement is not present
Rectangular Top Left, Bottom Right corners Lat,Lon coordinates	Coordinate System="LATLON" Top Left Corner=52.70,6.30 Bottom Right Corner=52.75,6.35	coordinates expressed in degree units the first coordinate is for lat, the second for lon the coordinate system statement must be present
Rectangular Top Right Bottom Left cor ners Row, Column coordinates	Coordinate System="ROWCOL" Top Right Corner=300,200 Bottom Left Corner=100,500	the first coordinate is for row, the second for column the row,col coordinate system is assumed by default if the Coor dinate System statement is not present
Rectangular Top Right, Bottom Left corners Lat,Lon coordinates	Coordinate System="LATLON" Top Left Corner=52.78,6.30 Bottom Right Corner=52.75,6.35	coordinates expressed in degree units the first coordinate is for lat, the second for lon the coordinate system statement must be present
Rectangular Center in row,col Size in pixel	Centre=200,300 Size Unit="ROWCOL" Size=100,150	the first coordinate is for row, the second for col the first size is for row, the second for col
Rectangular Center in row,col Size in km	Centre=50,50 Size Unit="KM" Size=1.5,2.0	the first size is for km in row direction1, the second size is for km in col direction
Rectangular Center in lat,lon Size in pixel	Coordinate System="LATLON" Centre=52.460,5.519 Size Unit="ROWCOL" Size=100,150	
Rectangular Center in lat,lon Size in km	Coordinate System="LATLON" Centre=52.460,5.519 Size Unit="KM" Size=1.5,2.0	
Polygonal Vertices in row,col	Coordinate System="ROWCOL" Number of Vertex=4 Ver tex=100,250,250,100,400,250,250, 400	the vertices are in the form V1row,V1col,V2row,V2col,... ,V nrow,Vncol the row,col coordinate system is assumed by default if the Coor dinate System statement is not present
Polygonal Vertices in lat,lon	Coordinate System="LATLON" Number of Vertex=4 Vertex=52.78,6.41,52.79,6.34, 52.76,6.32, 52.75,6.39	the vertices are in the form V1lat,V1lon,V2lat,V2lon,...,V nla t,Vnlon the coordinate system statement must be present

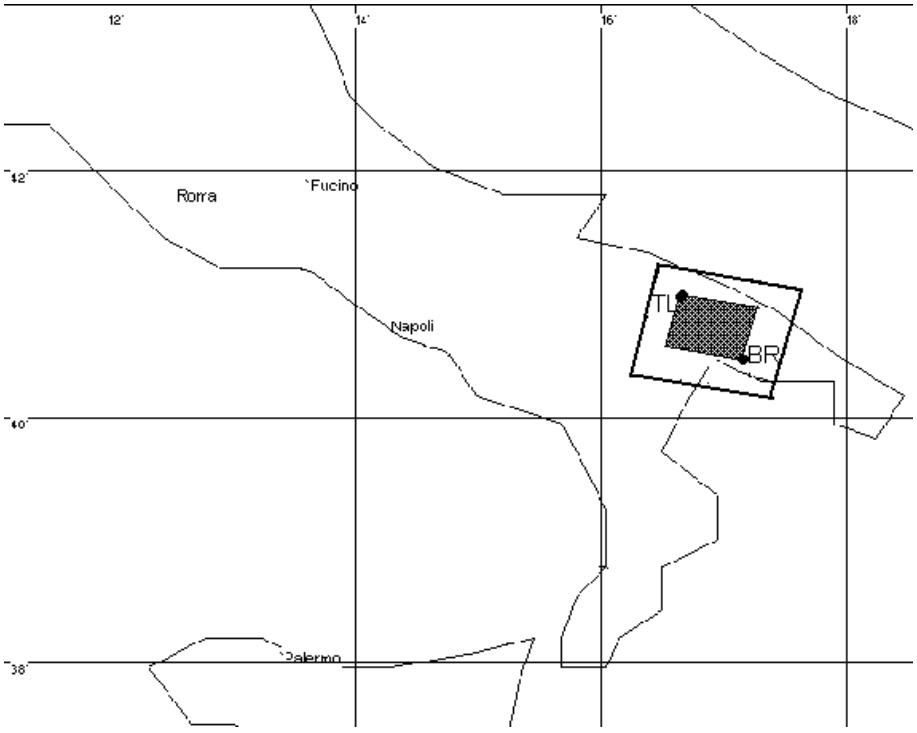
Some care must be taken when using an AOI specified through the Top Right and Bottom Left corners in Latitude, Longitude system, for products which are not geocoded (like the RAW, SLC or PRI). Lets look at a SAR image seen it in a geographical projection (see figure “PRI seen in a geographical chart”). When you select the two corners TL and BR, you are thinking of the AOI shown in “What the user was thinking” but due to the fact that the range axis is not parallel neither to the latitude neither to the longitude axes you could obtain the AOI shown in “What the user could obtain”. This happens because the system converts the two corners coordinates from lat,lon to row,col and then uses these to extract the related AOI.



PRI seen in a geographical chart.



What the user was thinking.



What the user could obtain.

To avoid these problems an “ad hoc” management of such situations has been developed. For all tools except the Statistical tool with the local statistic, the AOI obtained is the same as in the figure “what the user could obtain” (i.e. the rectangular area with sides parallel to the range and azimuth axes having the specified TL and BR corners) and a warning message is issued. Due to the capability of the statistical tool to manage polygonal regions, the situation shown in the

figure “what the user was thinking” does apply (notice that only a polygonal region can have sides not parallel to the range and azimuth axes).

AOI specification by example image

The remaining way to specify the AOI is associated to the possibility to give to the system not the AOI vertices in some coordinate system, but an example image. In other words, an AOI can be specified through an image portion, selected in some way. This approach is limited in the sense that is implemented only in the extraction tool and only for the quick look image. Starting from the quick look image, you can select a portion (using IDL or XV or another image processing system capable to crop an image) and tell to the system to retrieve the related coordinates. In this way you will obtain the Top Left, Bottom Right coordinates in row,col system, expressed in the full resolution reference, ready to be used in the remaining tools, without knowing anything about the coordinates, just seeing the image and visually selecting the AOI.

Appendix 5: Sequential Execution of Toolbox Algorithms

The SAR Toolbox tools can handle ".INI" files with multiple header sections in order to execute the various functions in a sequential way. Using this technique the creation of processing chains became a very simple task, requiring just the appending of the various elementary ".INI" files in one unique file and their subsequent activation. Let us suppose that we need to generate from a SLC image portion (already extracted) the related square modulus. This can be obtained with the data conversion tool first converting the complex input to modulus and then raising it to the square.

These two operations can be combined in the following unique ".INI" file (save it with name powmodulus.ini):

```
[COMPLEX TO AMPLITUDE]
Input Dir = "./"
Output Dir = "./"
Input Image = "slcimage.e5"
Output Image = "modulus"
[AMPLITUDE TO POWER]
Input Dir = "./"
Output Dir = "./"
Input Image = "modulus.v2"
Output Image = "power_modulus"
```

executed with the following command:

```
stbx powmodulus.ini
```

The stbx tools have another important feature. When the first section in the composite ".INI" file is [GLOBAL SETTING], the related parameters have a global meaning and can be used by all the functions invoked in the ".INI" file.
This section can contain assignments for the following parameters only:

- input directory
- output directory
- temporary directory
- delete input file flag

The previous example can then be transformed in:

```
[GLOBAL SETTING]
Input Dir = "./"
Output Dir = "./"
Temp Dir = "./"
Delete Input Image = 'Y'
[COMPLEX TO AMPLITUDE]
Input Image = "slcimage.e5"
Output Image = "modulus"
[AMPLITUDE TO POWER]
Input Image = "modulus.v2"
Output Image = "power_modulus"
```

If one of the global parameters is also inserted in one of the “.INI” sections, it overrides the global setting but just for such section.
In the following example:

```
[GLOBAL SETTING]
Input Dir = "./"
Output Dir = "./"
Temp Dir = "./"
Delete Input Image = 'Y'
[COMPLEX TO AMPLITUDE]
Temp Dir = "tmp"
Input Image = "slcimage.e5"
Output Image = "modulus"
[AMPLITUDE TO POWER]
Input Image = "modulus.v2"
Output Image = "power_modulus"
```

the temporary directory, globally set to "./", changes to tmp for the complex to amplitude conversion only.

Appendix 6: System Performance and Memory Issues

The parameters which affect the system performance of the SAR Toolbox system, on the various machines are described here. These parameters are all related to the main memory allowable to the SAR Toolbox. This amount, shall be carefully selected for multi user machines like the Sun or the DEC. A too small amount of such memory imply that the SAR Toolbox system dimension its image buffer to a very little extent so requiring many disk access to cover the image processing action involved. On the other hand, if this size exceeds the physical memory allowable for the process the OS began to swap the memory on disk, causing again loss of performances. For single user machines, like PC or Mac use the OS commands to measure the amount of free physical memory. If you are in doubt, select few Mbytes less than the amount of installed physical memory. The system ".INI" file is listed below and is the same for all the machines, so you have to modify just the section related to your computer. The memory amount is expressed in kbytes.

[SUN]
System Memory = 16384
[SGI]
System Memory = 16384
[HP]
System Memory = 16384
[OSF]
System Memory = 16384
[IBM]
System Memory = 16384
[DOS]
System Memory = 16384
[WIN95]
System Memory = 16384
[MAC]
System Memory = 16384

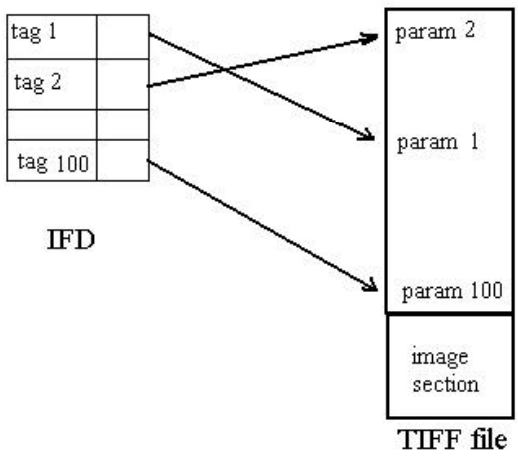
Appendix 7: The SAR Toolbox Internal Format

The internal format adopted in STB is called TTIFF which stands for Tiled Tagged Image File Format. The TTIFF format is a particular form of the TIFF format and is very similar to the internal format of the Italian PAF products (which is called BTIFF, Blocked Tagged Image File Format). The slight differences are essentially associated with the name of some image parameters (which, in the TIFF world, are called “tags”) and with some restrictions in the image organization. To clarify the topic, a brief description of TIFF, BTIFF and TTIFF formats follows:

The TIFF format is a image file format used mostly for the PC Desktop Publishing applications for encouraging the exchange of digital images between the various packages. The main features of the TIFF format are:

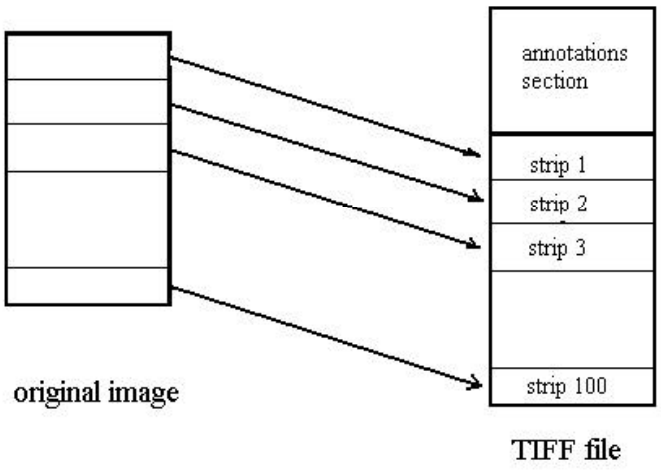
capable to describing bi-level, greyscale, palette-color and full RGB color image data in several color spaces
includes a number of image compression schemes
is not tied to specific hardware
is portable; it does not favor particular operating systems, file systems, compiler or processors
is designed to be expandable and evolve as new needs arise
allows the inclusion of an unlimited amount of private or special-purpose information
allows the presence of any number of images in one file, each with his own set of annotations

A TIFF file is logically divided in two sections: an annotations data part and an image data part. The annotations are stored and retrieved in a TIFF file by their name, i.e. the tag number. This number is used as a entry for a table (called IFD, Image File Directory) of pointers that show the zone of the TIFF file in which the annotations is kept. In this way a change of the position of the parameters (or the adding of further ones) does not affect the capability to retrieve the data (because all the read operations use this indexed mechanism). For these reasons the TIFF format has a high tolerance to the annotations evolution (position change, new fields, change of datatype for an annotation, and so on).



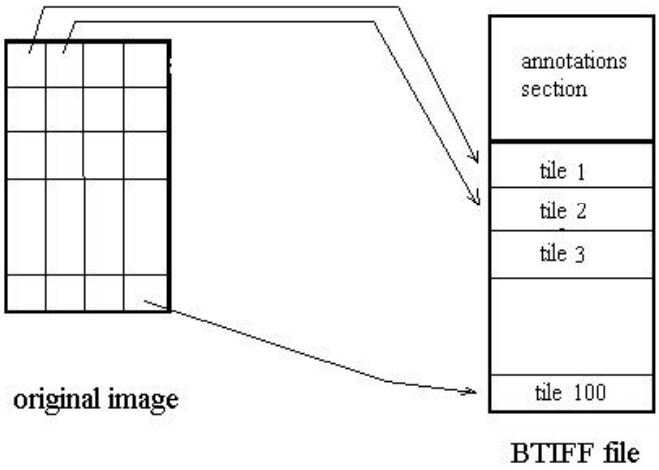
The pointers that show the zone of the TIFF file in which the annotations are kept.

The image section is kept in the TIFF as a sequence of strips that have the same columns as the original image and contains a number of rows (this number is chosen to obtain strips of a given size like 8 Kbytes, 16 Kbytes and so on). The location of the strips is stored in the same indexed way as for the image parameters. In this way the image can be easily accessed in subsections and moreover, when the compression is applied, a strip can be efficiently treated by the compression SW.



The image section is kept in the TIFF as a sequence of strips.

The BTIFF (Blocked TIFF) is an evolution of this format and has the following characteristics: the image is kept on file as tiles (instead of strips) to improve the image access efficiency independently of the access direction and the position of the sub-image accessed.



The BTIFF (Blocked TIFF).

The handling library does all the job and the user does not have to worry about this structure (he can read an image portion in the same way he reads an entire RASTER file). Like TIFF, the image and the parameters are kept in the same file and in a non positional format (i.e. an index

table is used). If the format or the number of the parameters is changed there is always a backward compatibility and moreover the handling library does not have to be changed (in this way a "dynamic" format can be handled). Because the BTIFF is a particular subset of the TIFF, the compatibility is maintained and it is possible to import and export images in TIFF with the same routines as the BTIFF one. The BTIFF has no limits in the handling of the multi-band data and/or having a pixel size different from 8 bit (e.g. complex images and so on). It is possible to compress the image via the LZW algorithm. The parameters can be maintained in their natural format e.g. the strings are kept as strings and the numbers as their binary representation (and NOT as strings too).
Efficiency in read-write operations with respect to the memory buffer usage is granted by tuning two parameters which controls the row and column dimensions of the tile, i.e. the elementary unit of the image. At IPAF we use a tile of 100 pixels 100 lines but different values can be used (and this is both transparent to the user and maintains the compatibility).

New TIFF specifications (TTIFF) were issued in parallel to the development of the BTIFF format. These are largely compatible, but differ at a few points:
the name of some parameters are different (e.g. BLOCKOFFSET has the tag 324 in the TTIFF and 273 in the BTIFF
the number of rows and columns of the tile are constrained to be a multiple of 16 for TTIFF (no limitations for the BTIFF)
in the BTIFF the image can be stored with the tiles organized horizontally or vertically (this feature does not exists in TTIFF)
Because the TTIFF is now a standard for many image processing packages (like XV for UNIX or ULEAD for PC) and because the handling library is exactly the same, the TTIFF has been selected as the internal format of STB. This will permit the direct ingestion of the STB images into display SW which accept as input the TTIFF format.
The unique limitation of TTIFF compared to BTIFF is the loss of the possibility to store the tiles vertically (which is of some utility only in the case of a vertical image elaboration, e.g. like the azimuth compression in SAR processing) but is surpassed by the advantage of the direct ingestion of the STB internal images in the image processing SW, without any conversion to standard TIFF.
However, to permit the visualization of the STB images under old SW which does not have the TTIFF ingestion capability, the STB will allow the generation of non tiled standard TIFF images with some little modifications both to the tags concerning the tile dimensions and a re-formatting of the image in order to transform tiles into strips.
In case of complex and other non 8-bit images a transform to a single 8bit image is performed.
IDL is capable of reading TIFF images using the *TIFF_READ* command having the following syntax:
Result = TIFF_READ(*File [, R, G, B]*)
where *File* is the file name of the image, *R, G* and *B* are optional vectors used to store the look-up table of a Palette color image and *Result* is a two-dimensional matrix containing the image pixels. If the TIFF image is a RGB true color one, *Result* will be a three-dimensional matrix holding in plane 0, 1 and 2 each one of the RGB components.
ERMAPPER includes an import menu to load a TIFF image and transform it into its internal format (see next paragraph). This option can be also activated via the operating system shell with the following command:
importmany *TIFF-Image-File ERMAPPER-Image-File*
The TIFF gray-level image files are transformed into a one band ERMAPPER file, while both RGB true color and Palette color images are always transformed into three band ERMAPPER image files.

Appendix 8: Further INI File Issues; setting values during run-time and using the “pipe” capability

The language for defining parameter values inside the INI file make it possible to set values run-time during the execution of tools. The usual rule

<parameter_name> <assign_symbol> <parameter_value>

where

<parameter name> is the name of the parameter to be set,

<assign> is the ‘=’ symbol,

and

<parameter_value> is “<string_value>” (for STRING and CHAR parameters) or <number_value> (for INTEGER and REAL parameters)

also allows the symbol ‘?’ to be used in place of the <string_value> or <number_value> terms.

So, when the SAR Toolbox encounters in an INI file a line like the following

Input Image = "?"

it suspends the execution, and prompts the user with the following question:

Enter a STRING value for Input Image :

The user has to supply a value (here a string, in accordance to the parameter type) which becomes the value of Input Image parameter for the execution of the task having that line.

The ? syntax is allowed also for non-string parameters as in the following sample line:

Constant Factor = ?

In this case, the SAR Toolbox suspends the execution and prompts the following question:

Enter a NUMBER value for Constant Factor :

A richer definition makes it possible to put after the ? symbol a help string (without blanks) to be used when constructing the question. This is useful especially for vector parameters. So, a line like the following

Top Left Corner = ?Row, ?Column

will give, during the SAR Toolbox execution, to two questions:

Enter a NUMBER value for Top Left Corner Row:
Enter a NUMBER value for Top Left Corner Column:

which are more clear than these two questions

Enter a NUMBER value for Top Left Corner :
Enter a NUMBER value for Top Left Corner :

which would be prompted in response to the following line, where no help strings have been inserted after the ? symbol.

Top Left Corner = ?, ?

The help string can also assist the user to supply a correct answer. The answer related to a “Yes/No” parameter to be supplied run-time can be defined using a line like the following:

Delete Input Image = "?(Y/N)"

The question prompted from the SAR Toolbox is the following:

Enter a CHAR value for Delete Input Image (Y/N):

An extensive use of the symbol ? inside the INI files allows to build more general files, both reusable and closer to program instructions.

The following two INI files can be used for applying the gain conversion module to an input image, and then passing the converted image (tmp.GCi) to TIFF format. At the end, the converted image is deleted.

[GAIN CONVERSION]
Input Image = "?"
Output Image = "tmp"
Min Percentage = ?
Max Percentage = ?
Number of Black Levels = 0.0

[TIFF GENERATION]
Delete Input Image = "Y"
Input Images = "tmp.GCi"
Output Image = "?"

The following text shows the related output obtained from its execution (bold style refers to user-supplied parameter values and “GC2TIFF.INI” is a text file containing both of the above “GAIN CONVERSION” and “TIFF GENERATION” INI file instructions):

$ stbx GC2TIFF.INI

SAR TOOLBOX: Generic Tool ver. 1.2

Doing GAIN CONVERSION

Enter a STRING value for Input Image : i09.XTs
Enter a NUMBER value for Min Percentage : 1.0
Enter a NUMBER value for Max Percentage : 99.0

stbx warning. No AOI parameters found. Get the whole image
- 95% completed.

GAIN CONVERSION completed.

Doing TIFF GENERATION

Enter a STRING value for Output Image : i09
- 95% completed.

TIFF GENERATION completed.

Exploiting the UNIX and DOS operating system “pipe” capability

Exploiting the UNIX and DOS operating system “pipe” capability it is possible to create files containing the answers related to the ? symbols inserted into a INI file and, so, further automate the execution.
If we consider the previous example, we may create a file containing several lines each one containing some input data, e.g. a file GC2TIFF.PAR, may contain the following:

```
i09.XTs
1.0
99.0
i09
```

The SAR Toolbox Generic Tool can be run using the following command

```
cat GC2TIFF.PAR | stbx GC2TIFF.INI
```

on UNIX machines, or the following command

```
type GC2TIFF.PAR | stbx GC2TIFF.INI
```

for PC machines.

Moreover, on UNIX machines, having a collection of input files each containing a different set of parameter values to be given as input to the same INI file, it is possible to create inside a C-shell a file like this

```
#/bin/csh
foreach m (GC2TIFF*.PAR)
cat $m | stbx GC2TIFF.INI
end
```

which, at execution time, will look at all the files GC2TIFF*.PAR and, for any file matching this pattern, run the SAR Toolbox Generic Tool using the value of the instanced variable m. This means that, having ten files GC2TIFF0.PAR, ... , GC2TIFF9.PAR, each created with appropriate data, in the directory where this C-shell file is executed, the SAR Toolbox is run ten times, each time with one of the different matched files. Using this technique will help to avoid the need for the intervention of the user.

A similar feature cannot be directly implemented using the DOS operating system. It is however possible to collect inside a Batch file, named e.g. GC2TIFF.BAT, all the sequences of SAR Toolbox activation command, as in the following example:

```
type GC2TIFF0.PAR | stbx GC2TIFF.INI
type GC2TIFF0.PAR | stbx GC2TIFF.INI
                  ...                ...
type GC2TIFF9.PAR | stbx GC2TIFF.INI
```

The FOR command of the DOS operating system may help to create this file. In fact, instead of extensively writing all these lines, it is possible to create the GC2TIFF.BAT running from the Prompt MS-DOS shell the following commands:

```
DEL GC2TIFF.BAT
FOR %m IN (GC2TIFF*.PAR) DO ECHO TYPE %m ! stbx GC2TIFF.INI >> GC2TIFF.BAT
```

The first command line is needed to be sure that redirection happens in empty file. The ! symbol, in the second command line, is used in place of | to suspend pipeline activation. At the end, it is only necessary edit the GC2TIFF.BAT file and search and replace all ! occurrence with the | symbol. Finally, to execute all command inside the GC2TIFF.BAT file is only needed to run from the DOS shell the following command

```
GC2TIFF.BAT
```

Finally, no kind of pipe capability exists on MacIntosh machines.